

# SENATI

SERVICIO NACIONAL DE ADiestramiento INDUSTRIAL

DIRECCIÓN ZONAL: LIMA - CALLAO

ESCUELA DE TECNOLOGIA DE LA INFORMACIÓN

CFP LUIS CACERES GRAZIANI

INGENIERIA DE SOFTWARE CON INTELIGENCIA ARTIFICIAL

Proyecto de Innovación y/o Mejora

Nivel Profesional Técnico / Técnico operativo

**DESARROLLO DE UN APLICATIVO MÓVIL CON SISTEMA  
DE RECOMENDACIÓN DE SALUD BASADO EN IA, PARA  
LA GESTIÓN DE CITAS MÉDICAS EN LA CLÍNICA RENÉ  
REHABILITACIÓN S.A.C.**

Autor:

Sebastian Huertas Chancos

Asesor:

Mg. Jose Armando Tiznado Ubillus

Lima - Perú

2025

## **Agradecimiento**

Agradezco a Dios por darme esta oportunidad y también agradezco a mis padres por su apoyo incondicional en estos 3 años de lucha, ya que cada ciclo para mí ha sido difícil de avanzar y mejorar. Continuamente agradezco a esta institución por orientarme a seguir mis sueños. También agradezco a mi profesor Mg. José Armando Tiznado Ubillus, ya que nos ayudó con esta tesis y nos motivó a tener como iniciativa este proyecto.

## **Dedicatoria**

Este proyecto se lo dedico a mis padres y a mi hermano ya que con su apoyo me hizo mejorar y reflexionar sobre la importancia de seguir estudiando. Pero, más que nada, dedico este proyecto a mí mismo, ya que fueron meses de lucha y sacrificio para mejorar cada día.

# **Resumen Ejecutivo del Proyecto de Innovación / Mejora / Creatividad**

El presente proyecto de innovación tecnológica tiene como finalidad el diseño, desarrollo e implementación de un aplicativo móvil orientado a mejorar de manera significativa la gestión de los servicios clínicos y la atención personalizada al paciente en la Clínica René Rehabilitación S.A.C. Esta iniciativa responde a la necesidad de modernizar los procesos administrativos, facilitar el acceso a servicios médicos y ofrecer un canal interactivo entre el paciente y el centro de salud.

La solución tecnológica desarrollada integra un sistema de recomendación de salud basado en inteligencia artificial (IA Gemini), que permite ofrecer sugerencias rápidas, de acuerdo a la recomendación que necesite cada paciente, contribuyendo así a la orientación preventiva y al seguimiento continuo de su estado de salud.

La aplicación cuenta con un módulo de autenticación segura y registro de usuarios, permitiendo la reserva digital de citas médicas, así como la visualización de las mismas en tiempo real. Asimismo, se ha desarrollado un panel administrativo dirigido al personal clínico y administrativo, desde el cual se puede gestionar la asignación de terapeutas, áreas y horarios de atención, y enviar confirmaciones de citas mediante SMS de forma automatizada.

Este proyecto no solo representa una innovación en la digitalización de los procesos clínicos, sino que incorpora elementos de inteligencia artificial como valor diferencial, permitiendo una atención más eficiente, accesible y centrada en el paciente. La solución desarrollada mejora la trazabilidad de la atención, reduce los errores humanos, disminuye el tiempo de espera, y contribuye a una mayor satisfacción de los usuarios, tanto internos como externos.

En un contexto donde la transformación digital en salud es prioritaria, esta iniciativa se presenta como un modelo replicable para otras instituciones médicas que buscan implementar herramientas inteligentes que integren automatización, eficiencia y humanización del servicio. Así, se consolida como una propuesta de alto impacto tecnológico, funcional y social.

# Índice general

<b>1. Generalidades de la Empresa</b>	<b>12</b>
1.1. Datos Generales de la Empresa . . . . .	12
1.1.1. Datos de la Empresa . . . . .	12
1.1.2. Razón Social . . . . .	13
1.1.3. Logo de la Empresa . . . . .	13
1.1.4. Localización geográfica . . . . .	14
1.2. Misión, Visión, Objetivo de la Empresa, Valores de la Empresa . . . . .	14
1.2.1. Misión . . . . .	14
1.2.2. Visión . . . . .	14
1.2.3. Objetivo . . . . .	14
1.2.4. Valores . . . . .	15
1.3. Productos, Mercado, Cliente . . . . .	15
1.3.1. Producto . . . . .	15
1.3.2. Mercado . . . . .	15
1.3.3. Cliente . . . . .	16
1.4. Estructura de la organización . . . . .	17
<b>2. Plan del proyecto de Innovación / Mejora / Creatividad</b>	<b>18</b>
2.1. Identificación del Problema Técnico . . . . .	18
2.1.1. Problemática Local . . . . .	18
2.1.2. Problemática Nacional . . . . .	18
2.1.3. Problemática Internacional . . . . .	19
2.1.4. Problemática General . . . . .	20
2.1.5. Problemáticas Específicas . . . . .	20
2.2. Objetivo del Proyecto de Innovación . . . . .	20
2.2.1. Objetivo General . . . . .	20
2.2.2. Objetivos Específicos . . . . .	20
2.3. Antecedentes del Proyecto de Innovación . . . . .	21
2.3.1. Antecedentes Nacional . . . . .	21
2.3.2. Antecedentes Internacional . . . . .	22
2.3.3. Antecedentes Artículo Científico . . . . .	23
2.4. Justificación del Proyecto de Innovación / Mejora / Creatividad . . . . .	24
2.4.1. Justificación Teórica . . . . .	24
2.4.2. Justificación Personal . . . . .	25
2.4.3. Justificación Práctica . . . . .	25
2.4.4. Justificación Tecnológica . . . . .	25
2.5. Marco Teórico y Conceptual . . . . .	28

2.5.1. Fundamentos Teóricos del Proyecto de Innovación / Mejora / Creatividad . . . . .	28
2.5.2. Bases Teóricas . . . . .	28
2.5.3. Tecnologías y Herramientas . . . . .	36
<b>3. Análisis de la Situación Actual</b>	<b>38</b>
3.1. Diagrama del Proceso, Mapa del Flujo de Valor y/o Diagrama de Operación Actual . . . . .	38
3.2. Efectos del Problema en el Área de Trabajo o en los Resultados de la Empresa . . . . .	39
3.3. Análisis de las Causas Raíz que Generan el Problema/Necesidad . . . . .	39
3.4. Priorización de Causas Raíz . . . . .	41
3.4.1. Lista de Causas Encuestadas . . . . .	41
3.4.2. Diagrama de Pareto en Función al Grado de la Causa . . . . .	42
<b>4. Propuesta Técnica de la Mejora</b>	<b>46</b>
4.1. Plan de acción de mejora Propuesta . . . . .	46
4.1.1. Fase 1: Iniciación . . . . .	47
4.1.2. Fase 2: Planificación . . . . .	51
4.1.3. Fase 3: Implementación . . . . .	58
4.2. Módulos Implementados . . . . .	62
4.2.1. Módulo 1: Generar la API de Gemini en Google AI Studio . . . . .	62
4.2.2. Módulo 2: Pantallas de Inicio, Login y Registro . . . . .	64
4.2.3. Módulo 3: Reserva y Visualización de Citas . . . . .	70
4.2.4. Módulo 4: Implementación de la IA con Gemini en Android Studio . . . . .	75
4.3. Implementación del Chat de Consultas con Gemini 1.5 Flash . . . . .	77
4.3.1. Creación de las clases e Interfaces para la consulta IA . . . . .	79
4.3.2. Módulo 5: Panel de Administración . . . . .	88
4.3.3. Arquitectura del sistema . . . . .	93
4.3.4. Cronograma de ejecución del Proyecto . . . . .	94
4.3.5. Diagrama de Proceso Mejorado . . . . .	95
<b>5. COSTO DE IMPLEMENTACIÓN DE LA MEJORA / INNOVACIÓN / CREATIVIDAD</b>	<b>97</b>
5.1. Costo de materiales . . . . .	97
5.2. Costo de mano de obra . . . . .	97
5.3. Costo de maquinaria, herramientas y equipos . . . . .	98
5.3.1. Costo de depreciación de herramientas . . . . .	98
5.4. Depreciación de las computadoras . . . . .	98
5.5. Otros costos de implementación de la mejora . . . . .	98
5.5.1. Costo de Servicios y Consumo de Luz . . . . .	98
5.5.2. Lista de gastos personales . . . . .	99
5.5.3. Costo total de la implementación de la mejora . . . . .	99
<b>6. EVALUACIÓN TÉCNICAS Y ECONÓMICAS DE LA MEJORA / INNOVACIÓN / CREATIVIDAD</b>	<b>100</b>
6.1. Beneficio técnico y / o económico esperado de la mejora / innovación / creatividad . . . . .	100
6.1.1. Relación Beneficio / Costo . . . . .	100

6.2.	Identificación de pérdida económica . . . . .	101
6.2.1.	Total promedio de pérdidas anuales . . . . .	102
6.2.2.	Total de la implementación del proyecto . . . . .	102
6.2.3.	Total promedio de pérdidas anuales después de la mejora . . . . .	102
6.2.4.	Relación beneficio/costo . . . . .	103
6.2.5.	Tiempo de recuperación de la inversión . . . . .	103
<b>7.</b>	<b>Conclusiones respecto a los objetivos del Proyecto de Mejora/Innovación/Creatividad</b>	<b>104</b>
7.1.	Resultados . . . . .	104
7.1.1.	Conclusión . . . . .	106
<b>8.</b>	<b>Recomendaciones para la empresa respecto del Proyecto de Mejora/Innovación/Creatividad</b>	<b>107</b>
8.1.	Recomendaciones . . . . .	107
<b>Capítulo</b>		<b>109</b>
<b>Referencias</b>		<b>112</b>

# Índice de figuras

1.1.	Logo representativo de RENE Neuromotora . . . . .	13
1.2.	Ubicación física de las instalaciones centrales . . . . .	14
1.3.	Organigrama de la Empresa . . . . .	17
3.1.	Organigrama de la Empresa . . . . .	38
3.2.	Diagrama de Ishikawa . . . . .	40
3.3.	Diagrama de Pareto 1 . . . . .	42
3.4.	Diagrama de Pareto 2 . . . . .	42
3.5.	Gráfico de la causa I: Falta de herramientas automatizadas . . . . .	43
3.6.	Gráfico de la causa II: Recomendaciones no personalizadas . . . . .	44
3.7.	Gráfico estadístico de la causa III: Dependencia de datos subjetivos . . . . .	45
4.1.	Crear clave de API . . . . .	62
4.2.	Crear clave de API . . . . .	63
4.3.	Pantalla de Inicio . . . . .	64
4.4.	Interfaz de bienvenidaActivity . . . . .	64
4.5.	Pantalla de Inicio . . . . .	65
4.6.	Clase LoginActivity . . . . .	66
4.7.	Pantalla de Login . . . . .	67
4.8.	Clase RegisterActivity . . . . .	68
4.9.	Pantalla de Registro . . . . .	69
4.10.	Clase ReservaCitaActivity . . . . .	70
4.11.	Pantalla para reservar una cita . . . . .	71
4.12.	Clase CitaAdapter . . . . .	72
4.13.	Item de Lista de Citas . . . . .	72
4.14.	Visualización de citas registradas . . . . .	73
4.15.	CardView de Paciente . . . . .	74
4.16.	Código del BakingActivity . . . . .	79
4.17.	Código del BackingScreen . . . . .	80
4.18.	Código del BakingViewModel . . . . .	81
4.19.	Pantalla de consulta médica con IA . . . . .	82
4.20.	Código de UI . . . . .	83
4.21.	Pantalla de consulta médica con IA . . . . .	84
4.22.	Pantalla de consulta médica con IA . . . . .	86
4.23.	Nota: Elaboración Propia. . . . .	87
4.24.	Interfaz de AdminActivity . . . . .	88
4.25.	Interfaz de AdminCitasActivity . . . . .	89
4.26.	Panel de administración de citas . . . . .	90
4.27.	Selector de Fecha y Hora . . . . .	91

4.28. Pantalla para asignar cita . . . . .	92
4.29. Nota: Elaboración propia . . . . .	93
4.30. Actividades del proyecto . . . . .	94
4.31. Diagrama de mejora . . . . .	95
4.32. Diagrama de Gantt . . . . .	96

# Índice de cuadros

1.1. <i>Información de la razón social de la empresa</i> . . . . .	13
3.1. <i>Lluvia de ideas de las causas</i> . . . . .	39
3.2. Identificación de Problemas según encuestas . . . . .	41
4.1. Tabla de visión del proyecto . . . . .	48
4.2. <i>Identificación de cargos y responsables</i> . . . . .	49
4.3. <i>Equipo de desarrollo del proyecto</i> . . . . .	49
4.4. <i>Requerimientos funcionales del sistema</i> . . . . .	50
4.5. <i>Requerimientos no funcionales del sistema</i> . . . . .	51
4.6. Tabla del Product Backlog . . . . .	52
4.7. <i>Matriz de impacto de prioridades</i> . . . . .	52
4.8. <i>Etiquetas para la clasificación del desarrollo del proyecto</i> . . . . .	52
4.9. <b>Sprint para el seguimiento del proyecto (Parte 1)</b> . . . . .	56
4.10. <b>Sprint para el seguimiento del proyecto (Parte 2)</b> . . . . .	57
5.1. <i>Costos estimados de materiales utilizados para la implementación del proyecto.</i> . . . . .	97
5.2. <i>Tabla de costos de mano de obra estimados para el desarrollo del proyecto.</i> . . . . .	97
5.3. <i>Detalle técnico y económico del equipo Laptop 01 utilizado en el desarrollo del proyecto.</i> . . . . .	98
5.4. <i>Cálculo de depreciación anual</i> . . . . .	98
5.5. <i>Cálculo de depreciación anual del equipo durante su vida útil estimada de 5 años.</i> . . . . .	98
5.6. <i>Lista de gastos personales del consumo de servicio de luz e internet</i> . . . . .	99
5.7. <i>Costos adicionales de luz e internet</i> . . . . .	99
5.8. <i>Total de la implementación del proyecto</i> . . . . .	99
6.1. Pérdidas económicas por falta de herramientas automatizadas para seguimiento remoto . . . . .	101
6.2. Pérdidas económicas por baja adherencia a ejercicios en casa . . . . .	102
6.3. Total promedio de pérdidas anuales . . . . .	102
6.4. Costo y eficiencia de la implementación del proyecto . . . . .	102
6.5. Comparación de pérdidas antes y después de la mejora . . . . .	102
6.6. Relación beneficio/costo . . . . .	103
6.7. Comparación de pérdidas antes y después de la mejora . . . . .	103

# Introducción

En la actualidad, los sistemas digitales han transformado significativamente la manera en que las instituciones de salud operan, mejorando la interacción con los pacientes, optimizando los procesos internos y facilitando el acceso a la información médica. Este cambio es impulsado principalmente por la digitalización, el uso de aplicaciones móviles y el aprovechamiento de tecnologías emergentes como la inteligencia artificial (IA). En este contexto, la integración de herramientas inteligentes como los chatbots, el procesamiento automático de datos clínicos y los sistemas de notificaciones personalizadas se han convertido en elementos fundamentales para elevar la calidad del servicio médico.

El presente proyecto tiene como finalidad el desarrollo e implementación de un sistema digital integral para la Clínica René Rehabilitación S.A.C., especializada en el tratamiento y rehabilitación neurológica. El sistema consiste en una aplicación móvil multiplataforma y un panel web administrativo, los cuales permiten una gestión eficiente de las citas médicas, así como una atención más personalizada y automatizada para los pacientes. Entre las funcionalidades destacadas se encuentra un chatbot inteligente basado en IA, que brinda respuestas orientativas sobre salud, así como notificaciones automáticas por mensajería SMS, permitiendo mantener informado al usuario sobre sus citas y tratamientos.

El sistema también incluye un módulo de autenticación de usuarios mediante Firebase, así como una base de datos en tiempo real para almacenar registros clínicos y administrativos. El uso de tecnologías modernas como Android Studio, Firebase, Gemini AI y plataformas web responsivas asegura una solución robusta, escalable y alineada con los desafíos actuales del sector salud.

A continuación, se presenta un resumen de lo que se encontrará en cada capítulo de este trabajo:

- **Capítulo 1: Generalidades de la Empresa**

Se describe a la Clínica René Rehabilitación S.A.C., su historia, misión, visión, objetivos, valores institucionales, servicios que brinda, su ubicación actual, el mercado al que se dirige y el organigrama de funcionamiento.

- **Capítulo 2: Plan de Proyecto de Innovación**

Se define el problema principal que motivó la creación del proyecto. Se establecen los objetivos generales y específicos, los antecedentes nacionales e internacionales relevantes, y el marco teórico necesario para sustentar el proyecto. Además, se describen las herramientas tecnológicas utilizadas, como Android Studio, Firebase y Gemini AI.

- **Capítulo 3: Análisis de la Situación Actual**

Se detalla el proceso actual de gestión de citas y atención en la clínica, identificando las debilidades y dificultades que afectan la eficiencia del servicio. Se analizan las causas de la problemática y su impacto en la atención al paciente.

- **Capítulo 4: Propuesta Técnica de la Mejora**

Se presenta la arquitectura del sistema, los requerimientos funcionales y no funcionales, el diseño de la interfaz móvil y web, y el uso de metodologías ágiles como el tablero Kanban. Se describen los recursos tecnológicos empleados y las limitaciones del sistema.

- **Capítulo 5: Costo de Implementación del Proyecto**

Se detalla el presupuesto necesario para la implementación del sistema, incluyendo costos de desarrollo, licencias, servicios de hosting, herramientas, así como recursos como mano de obra y servicios básicos.

- **Capítulo 6: Evaluación Técnica y Económica del Proyecto**

Se analizan los beneficios técnicos esperados, como la reducción del tiempo de atención, y los beneficios económicos, como el ahorro en procesos administrativos. Se calcula la relación costo-beneficio y el tiempo estimado de retorno de la inversión.

- **Capítulo 7: Conclusiones**

Se presentan las principales conclusiones del proyecto, destacando los resultados alcanzados, la mejora en la atención al paciente y la viabilidad de extender el sistema a otras áreas de la clínica.

- **Capítulo 8: Recomendaciones**

Se ofrecen sugerencias para futuras mejoras, como la implementación de autenticación por reconocimiento facial, la integración con historiales médicos electrónicos y el monitoreo de métricas de salud del paciente en tiempo real.

Además, el documento incluye los siguientes anexos:

- **Anexo 1:** Lista de problemas identificados en los procesos actuales de la clínica.
- **Anexo 2:** Matriz de operacionalización de variables: conceptos clave del sistema como eficiencia, accesibilidad y automatización.
- **Anexo 3:** Matriz de consistencia entre problema, objetivos, teorías aplicadas y variables del sistema.
- **Anexo 4:** Capturas de interfaz de la aplicación móvil y panel web.
- **Anexo 5:** Acta de constitución del proyecto.

# Capítulo 1

## Generalidades de la Empresa

### 1.1. Datos Generales de la Empresa

#### 1.1.1. Datos de la Empresa

- **RUC:** 20549480269
- **Nombre de la empresa:** RENE REEDUCACION NEUROMOTORA S.A.C.
- **Tipo:** Sociedad Anonima Cerrada
- **Dirección:** Av. Arequipa Nro. 3960, Miraflores
- **Correo electrónico del empresario:** miguel.cerna.rene@gmail.com
- **Empresario de la empresa:** Miguel Ángel Cerna Arrunátegui
- **Teléfono de la empresa:** +51 942 791 294
- **Estado:** Activo

#### 1.1.1.1. 1.1.2 Reseña Histórica

La Clínica René Rehabilitación S.A.C. fue fundada con la misión de brindar un servicio especializado en neurorrehabilitación, enfocado en mejorar la calidad de vida de pacientes que han sufrido lesiones neurológicas, accidentes cerebrovasculares, traumatismos, o padecen enfermedades degenerativas. Desde sus inicios, la clínica se ha caracterizado por su vocación de servicio, innovación y atención centrada en el paciente.

Con el paso del tiempo, nuestra institución ha evolucionado integrando nuevas tecnologías, desarrollando programas de terapias intensivas multidisciplinarias y ampliando sus servicios hacia la atención domiciliaria, reconociendo la importancia del entorno familiar en el proceso de recuperación.

La combinación entre la investigación médica, la formación continua de nuestro personal y el uso de herramientas tecnológicas ha posicionado a la clínica como una referencia en el ámbito de la rehabilitación neurológica en Lima.

Hoy, seguimos creciendo con el mismo espíritu que nos vio nacer: una actitud positiva, vocación de servicio y compromiso con la salud neurológica, apostando por un futuro donde cada paciente pueda recuperar su funcionalidad e independencia.

### **1.1.1.2. 1.1.3 ¿Quiénes somos?**

En Clínica René Rehabilitación S.A.C. somos un centro de salud especializado en neurorrehabilitación intensiva, comprometido con el bienestar integral de nuestros pacientes. Nos enfocamos en brindar una atención de calidad en el ámbito de la medicina física y la rehabilitación neurológica, combinando tecnología de vanguardia, terapias intensivas personalizadas y un enfoque humano basado en la investigación clínica.

Ofrecemos servicios tanto en nuestro centro en Miraflores como en la comodidad del hogar del paciente, lo cual nos permite adaptarnos a las necesidades reales de cada persona y su familia. Nuestro equipo de especialistas mantiene siempre una actitud positiva, empática y profesional, generando un ambiente de confianza y motivación para la recuperación.

### **1.1.2. Razón Social**

Cuadro 1.1: *Información de la razón social de la empresa*

Nombre de la empresa	Rene Neuromotora	Empresario	Miguel Cerna
Razón social	20549480269		
Dirección	Av. Arequipa 3960	<b>Distrito</b>	Miraflores
Rúbrica	Vendedor Comercial		

### **1.1.3. Logo de la Empresa**



Figura 1.1: Logo representativo de RENE Neuromotora

**Nota:** El diseño corporativo refleja la identidad visual de la organización.

#### 1.1.4. Localización geográfica

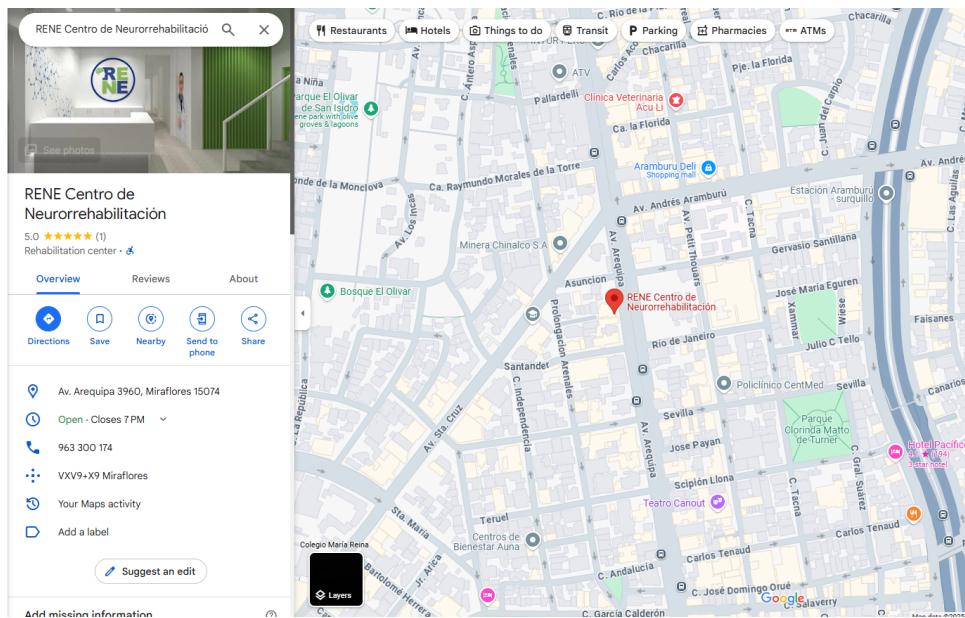


Figura 1.2: Ubicación física de las instalaciones centrales

**Nota:** Las oficinas principales se encuentran en Av. Arequipa 3960, Miraflores, Lima.

### 1.2. Misión, Visión, Objetivo de la Empresa, Valores de la Empresa

#### 1.2.1. Misión

Nuestra misión es brindar servicios de terapia física y rehabilitación neuromotora de alta calidad, integrando tecnologías innovadoras que optimicen el proceso terapéutico. Nos comprometemos a proporcionar una atención integral, personalizada y basada en evidencia, mediante el uso de software especializado para el seguimiento clínico, la evaluación del progreso y la generación automática de planes de tratamiento adaptados a cada paciente. Buscamos ser un soporte esencial en la recuperación funcional de nuestros pacientes, promoviendo su bienestar físico, emocional y social.

#### 1.2.2. Visión

Aspiramos a consolidarnos como la empresa líder en reeducación neuromotora en el país y en la región, reconocida por su enfoque innovador y humano. Nuestra visión es posicionarnos como referente de excelencia en rehabilitación neurológica, destacando por el uso de plataformas digitales inteligentes, un equipo altamente capacitado y un compromiso continuo con la calidad, la investigación y la mejora constante de nuestros servicios.

#### 1.2.3. Objetivo

Desarrollar e implementar un sistema informático integral que permita a la Clíni-

ca René Rehabilitación S.A.C. gestionar eficientemente la información de los pacientes, programar sesiones terapéuticas, monitorear el avance clínico y ofrecer tratamientos personalizados a través de algoritmos basados en datos. Este sistema tiene como objetivo mejorar la eficacia terapéutica, facilitar la comunicación entre profesionales de la salud y aumentar la satisfacción del paciente y su entorno familiar.

#### 1.2.4. Valores

**Compromiso:** Dedicamos todos nuestros recursos humanos y tecnológicos al bienestar de los pacientes, con responsabilidad y vocación de servicio.

**Innovación:** Apostamos por el uso constante de herramientas tecnológicas avanzadas, como inteligencia artificial, sensores biomédicos y aplicaciones móviles, para optimizar nuestros servicios.

**Calidad:** Cada procedimiento terapéutico y decisión clínica está respaldado por datos y análisis, garantizando resultados efectivos y medibles.

**Empatía:** Entendemos las necesidades físicas y emocionales de cada paciente, ofreciendo una atención personalizada y humanizada.

**Responsabilidad:** Actuamos con ética profesional, transparencia y respeto hacia los pacientes, sus familias y el equipo multidisciplinario.

### 1.3. Productos, Mercado, Cliente

#### 1.3.1. Producto

RENE REEDUCACIÓN NEUROMOTORA S.A.C. ofrece como producto principal servicios especializados de rehabilitación neuromotora. Estos servicios incluyen terapias físicas y ocupacionales personalizadas, enfocadas en la recuperación del movimiento y la funcionalidad del paciente. Además, la empresa incorpora tecnologías innovadoras, como inteligencia artificial, aplicaciones móviles y sensores biomédicos, para el seguimiento del estado del paciente, el ajuste automático de planes terapéuticos y la generación de recomendaciones clínicas personalizadas. Estas soluciones permiten mejorar la calidad del tratamiento y brindar una atención más eficiente y centrada en el paciente.

#### 1.3.2. Mercado

La empresa se desenvuelve en el sector salud, con enfoque en la rehabilitación física y neuromotora. Su mercado objetivo está compuesto por:

- Personas con lesiones neurológicas (ACV, parálisis, traumatismos, enfermedades degenerativas).
- Pacientes pediátricos con trastornos del desarrollo motor.
- Adultos mayores con pérdida de movilidad.
- Instituciones médicas (clínicas, hospitales, centros de salud) que necesitan soporte especializado o desean derivar pacientes a terapias avanzadas.

El uso de tecnologías modernas y un sistema informático especializado permite a la empresa diferenciarse en el mercado al ofrecer una atención más precisa, eficiente y continua.

### **1.3.3. Cliente**

Los principales clientes de RENE REEDUCACIÓN NEUROMOTORA S.A.C. son los pacientes que necesitan procesos de rehabilitación para recuperar su funcionalidad física y mejorar su calidad de vida. También se consideran clientes los familiares y cuidadores que participan activamente en el tratamiento y buscan soluciones confiables. Asimismo, los profesionales de la salud, como fisioterapeutas y médicos rehabilitadores, son usuarios clave de las herramientas tecnológicas que ofrece la empresa para realizar seguimiento clínico y tomar decisiones basadas en datos. Finalmente, también se consideran clientes a las instituciones médicas que contratan o derivan estos servicios terapéuticos.

## 1.4. Estructura de la organización

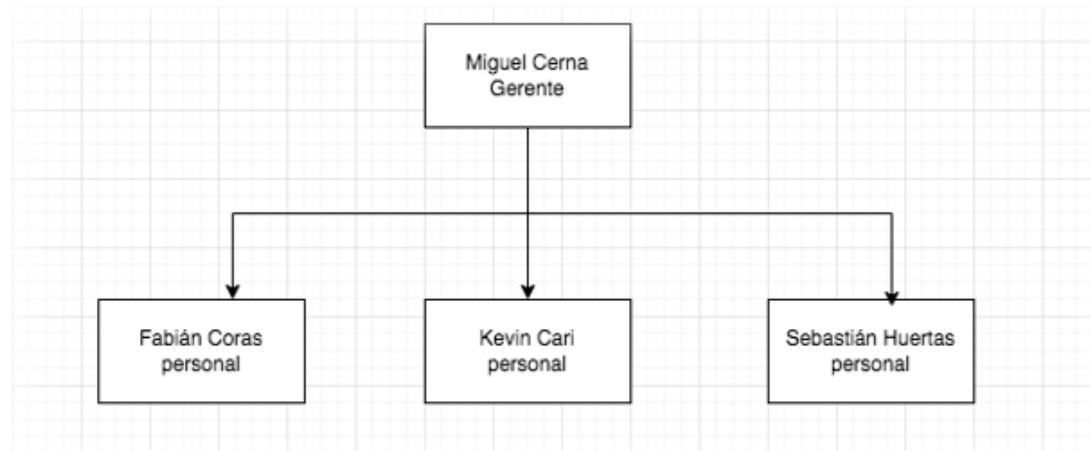


Figura 1.3: Organigrama de la Empresa

**Nota:** Se proyecta la estructura general del cargo ejecutivo de la empresa.

# Capítulo 2

## Plan del proyecto de Innovación / Mejora / Creatividad

### 2.1. Identificación del Problema Técnico

#### 2.1.1. Problemática Local

La empresa **RENE Reeducación Neuromotora S.A.C.**, especializada en brindar tratamientos de rehabilitación neuromotora, enfrenta actualmente dificultades importantes en el seguimiento continuo y personalizado del estado de salud de sus pacientes, especialmente una vez que estos abandonan el entorno clínico.

Durante las sesiones presenciales, los terapeutas logran evaluar el desempeño y evolución del paciente. Sin embargo, al concluir las sesiones, el control sobre su progreso se vuelve limitado o inexistente. Esto da lugar a una brecha entre el tratamiento aplicado en la clínica y el seguimiento domiciliario, lo cual compromete la eficacia del proceso terapéutico.

Muchos pacientes no ejecutan correctamente las rutinas recomendadas en casa, ya sea por olvido, falta de guía clara o desconocimiento sobre si están realizando los ejercicios de forma adecuada. En consecuencia, el avance clínico se ve afectado negativamente. Además, los terapeutas suelen depender del relato subjetivo del paciente para conocer su evolución fuera de consulta, lo cual carece de objetividad clínica.

A esto se suma que las recomendaciones actuales suelen ser generales y no se adaptan a los síntomas que el paciente pueda experimentar fuera del horario terapéutico. No se cuenta con una herramienta que interprete sus necesidades en tiempo real ni que brinde orientación automatizada y confiable sobre cómo proceder ante molestias, como rigidez muscular, fatiga o dolor leve.

*La problemática principal radica en la ausencia de un sistema tecnológico inteligente que permita acompañar al paciente fuera del entorno clínico, brindando recomendaciones personalizadas en tiempo real, lo que limita la continuidad del tratamiento y pone en riesgo una recuperación completa y sostenida.*

#### 2.1.2. Problemática Nacional

Según Giannina, (2019), en su tesis titulada “*Implementación de un sistema de recomendación de medicamentos en base a la naturaleza del paciente geriátrico*”, el objetivo

principal del proyecto es diseñar e implementar un sistema de recomendación enfocado en pacientes adultos mayores, considerando las características específicas de este grupo etario, como condiciones médicas previas, historial de tratamientos y posibles contraindicaciones.

A través del desarrollo de un prototipo funcional, la autora propone un enfoque que combina el análisis de datos clínicos con técnicas de recomendación para apoyar al personal médico en la selección de medicamentos adecuados.

De manera transversal, se abordan aspectos como la estructuración de la base de conocimiento, el perfil del paciente y la validación del sistema mediante pruebas con especialistas del sector salud.

### **2.1.3. Problemática Internacional**

Según De Croon et al., (2021), en su artículo científico titulado “*Health Recommender Systems: Systematic Review*”, el objetivo principal de esta investigación es realizar una revisión sistemática sobre los sistemas de recomendación aplicados al ámbito de la salud, evaluando su capacidad para influir positivamente en la toma de decisiones médicas y en la modificación de comportamientos relacionados con el bienestar del usuario .

A través del análisis de diversos estudios previos, los autores examinan el diseño, funcionamiento y efectividad de estos sistemas, así como su aplicabilidad en contextos reales de atención médica.

De manera transversal, se abordan aspectos clave como la personalización de las recomendaciones, la calidad de los datos utilizados, la transparencia de los algoritmos y los desafíos éticos relacionados con la privacidad de la información sensible del paciente.

## Formulacion del problema

### 2.1.4. Problemática General

¿Cómo mejorar el seguimiento remoto y la personalización del tratamiento de rehabilitación neuromotora de los pacientes de la Clínica René Reeducación Neuromotora S.A.C., mediante un sistema inteligente que genere recomendaciones personalizadas y emita alertas automáticas?

### 2.1.5. Problemáticas Específicas

- ¿Cómo permite el sistema recolectar información del paciente desde casa y agendar una cita sin que tenga que acudir a la clínica?
- ¿Cómo puede la integración de la IA de Gemini contribuir a generar recomendaciones terapéuticas automáticas y personalizadas, en función del progreso individual?
- ¿Qué impacto tendría la implementación de alertas automáticas y asignación administrativa de terapeutas sobre la eficiencia del tratamiento y la organización interna del centro?

## 2.2. Objetivo del Proyecto de Innovación

### 2.2.1. Objetivo General

Desarrollar e implementar un sistema inteligente para la Clínica René Reeducación Neuromotora S.A.C., que permita mejorar el seguimiento, la atención y la personalización del tratamiento de rehabilitación neuromotora fuera del entorno clínico, mediante una aplicación móvil conectada a un panel administrativo y a la API de Gemini para brindar recomendaciones clínicas útiles a los pacientes de forma automatizada, precisa y adaptada a sus necesidades.

### 2.2.2. Objetivos Específicos

- Facilitar el seguimiento del estado de salud del paciente desde su domicilio, mediante el registro remoto de síntomas, avances y actividades realizadas.
- Reducir el tiempo y la carga operativa del personal administrativo mediante un sistema automatizado de agendamiento de citas y asignación de terapeutas.
- Disminuir el número de dudas frecuentes gracias a la integración de un sistema de recomendaciones inteligentes a través de la API de Gemini, que brinda respuestas clínicas orientativas en función de los registros del paciente.
- Mejorar la adherencia al tratamiento a través de notificaciones personalizadas y recordatorios automáticos dentro de la aplicación móvil.
- Optimizar la calidad del soporte clínico remoto sin necesidad de intervención constante del personal terapéutico, gracias a la supervisión sistemática desde el panel administrativo.

- Aumentar la eficiencia del proceso de rehabilitación mediante la disponibilidad de información clínica objetiva y actualizada, proveniente del paciente en tiempo real.

## 2.3. Antecedentes del Proyecto de Innovación

### 2.3.1. Antecedentes Nacional

Según Giannina, (2019), la tesis desarrolla un sistema especializado en la recomendación de medicamentos para pacientes geriátricos, abordando un problema crítico en este grupo vulnerable: la polifarmacia y sus riesgos asociados. El estudio parte del análisis de las particularidades fisiológicas de los adultos mayores, como la disminución de la función renal y hepática, que afectan el metabolismo de los fármacos. El sistema propuesto integra una base de conocimiento farmacológico que incluye interacciones medicamentosas, contraindicaciones y ajustes de dosis recomendados para esta población. Utiliza un motor de reglas clínicas basado en estándares como los criterios Beers, permitiendo a los médicos ingresar las condiciones del paciente y recibir recomendaciones personalizadas que minimicen riesgos. La implementación incluye una interfaz amigable para el personal médico y se valida mediante casos de estudio reales, demostrando su potencial para reducir prescripciones inadecuadas en entornos clínicos geriátricos.

Según Francisco, (2020), la investigación se centra en el desarrollo de un sistema de recomendación médica innovador que utiliza bases de datos de grafos mediante Neo4j. El trabajo destaca cómo este enfoque supera las limitaciones de las bases de datos relacionales tradicionales al modelar eficientemente las complejas relaciones entre entidades médicas. El autor construye un grafo de conocimiento que conecta síntomas, enfermedades, exámenes diagnósticos y tratamientos, donde cada nodo y relación tiene propiedades específicas. El sistema implementa algoritmos de recorrido de grafos para realizar consultas como “dado este conjunto de síntomas, ¿cuáles son los diagnósticos más probables y sus tratamientos asociados?”. La tesis incluye un análisis comparativo que demuestra cómo este enfoque mejora la velocidad y precisión de las recomendaciones clínicas frente a sistemas tradicionales, particularmente en casos de diagnósticos complejos con múltiples comorbilidades.

Según Arturo, (2022), el trabajo presenta un sistema integral para la gestión automatizada de planes nutricionales en pacientes con enfermedades metabólicas. La investigación comienza con un análisis detallado de los modelos de decisión aplicables en nutrición clínica, seleccionando un enfoque híbrido que combina árboles de decisión con lógica difusa para manejar la incertidumbre en los datos de los pacientes. El sistema desarrollado permite ingresar parámetros como peso, talla, niveles de glucosa en sangre, alergias alimentarias y preferencias culturales, generando planes de alimentación personalizados con recomendaciones específicas de macronutrientes y micronutrientes. Incluye un módulo de seguimiento que analiza la adherencia al plan mediante registros alimentarios y marcadores bioquímicos, ajustando automáticamente las recomendaciones. La validación se realiza con pacientes diabéticos, mostrando mejoras significativas en el control glucémico y la calidad de vida.

Según Yesenia, (2020), la tesis explora a profundidad la aplicación de técnicas de inteligencia artificial en el proceso diagnóstico, con un enfoque particular en la medici-

na preventiva. La investigación compara diversos algoritmos de *machine learning* (redes neuronales convolucionales, máquinas de soporte vectorial y bosques aleatorios) aplicados a diferentes modalidades diagnósticas, incluyendo el procesamiento de imágenes médicas (radiografías y resonancias) y el análisis de datos clínicos estructurados. El estudio desarrolla un marco metodológico para evaluar la calidad de los datos de entrenamiento y optimizar los hiperparámetros de los modelos. Los resultados demuestran cómo estos sistemas pueden alcanzar una precisión comparable a especialistas humanos en ciertas patologías, al tiempo que identifica limitaciones críticas como el efecto de sesgos en los datos de entrenamiento. La autora propone un protocolo para la implementación ética y segura de estos sistemas en entornos clínicos reales.

Según Víctor, (2023), la investigación aborda el problema de la anemia infantil en contextos de recursos limitados mediante el desarrollo de una aplicación web predictiva. El estudio recopila un amplio conjunto de datos de pacientes pediátricos, incluyendo variables nutricionales, socioeconómicas, parasitológicas y de respuesta a tratamientos previos. Tras un exhaustivo proceso de limpieza y selección de características, se entrena y comparan varios modelos de *machine learning*, seleccionando un ensamblaje de *Gradient Boosting* por su equilibrio entre precisión e interpretabilidad. La aplicación resultante permite a los trabajadores de salud ingresar los datos del paciente y recibir inmediatamente una predicción de la probabilidad de éxito para diferentes protocolos de tratamiento, junto con recomendaciones personalizadas. El sistema se implementa en un centro de salud real, donde se evalúa no solo su precisión técnica sino también su usabilidad y aceptación por parte del personal médico. Los resultados muestran una reducción significativa en el tiempo requerido para alcanzar niveles adecuados de hemoglobina en los pacientes.

### 2.3.2. Antecedentes Internacional

Según Celia, (2022), la investigación se centra en el uso de inteligencia artificial (IA) para fortalecer los sistemas de salud en Chile, en línea con el Objetivo de Desarrollo Sostenible (ODS) 3 de la ONU, que busca garantizar una vida sana y promover el bienestar. El estudio analiza cómo las tecnologías de IA pueden optimizar la prestación de servicios médicos, mejorar la eficiencia en diagnósticos y tratamientos, y reducir inequidades en el acceso a la salud. La autora explora marcos éticos y regulatorios para implementar IA de manera confiable, destacando casos de uso como la predicción de brotes epidemiológicos, la gestión de recursos hospitalarios y la medicina personalizada. El trabajo concluye con recomendaciones políticas para integrar estas tecnologías en el sistema público de salud chileno, asegurando que sean transparentes, equitativas y alineadas con las necesidades de la población.

Según Matheus, (2021), la tesis aborda el desafío de la adherencia terapéutica mediante enfoques de *machine learning*. El autor desarrolla modelos predictivos para identificar pacientes con alto riesgo de incumplir tratamientos médicos, utilizando datos demográficos, clínicos y comportamentales. La investigación compara algoritmos como *Random Forest* y *XGBoost*, integrando también variables contextuales (como nivel socioeconómico y acceso a servicios de salud) para mejorar la precisión. Los resultados permiten estratificar pacientes y diseñar intervenciones personalizadas, como recordatorios automatizados o seguimiento intensivo para grupos de alto riesgo. El estudio destaca el potencial de

estas herramientas para reducir costos asociados al abandono terapéutico, especialmente en enfermedades crónicas como diabetes o hipertensión.

Según Vale, (2021), el trabajo propone un sistema de apoyo a decisiones basado en evidencia para mejorar la experiencia del paciente en hospitales de Qatar. Combina análisis de datos masivos (como satisfacción reportada por pacientes y tiempos de espera) con modelos de IA para identificar áreas críticas de mejora en la atención. La metodología incluye minería de texto en *feedback* de usuarios y técnicas de procesamiento de lenguaje natural (*NLP*) para categorizar quejas y sugerencias. El sistema genera *dashboards* interactivos para directivos hospitalarios, permitiendo priorizar acciones como redistribución de personal o optimización de flujos en urgencias. La validación en entornos reales demostró reducciones significativas en quejas por mala comunicación médico-paciente y retrasos en procedimientos.

Según Stubblefield, (2021), la investigación profundiza en algoritmos de IA aplicados a imágenes médicas y gestión sanitaria. El autor evalúa arquitecturas avanzadas de *deep learning* (como redes neuronales 3D y *transformers*) para tareas específicas: detección temprana de tumores en mamografías, clasificación de lesiones en dermatoscopías y predicción de readmisiones hospitalarias. Un aporte clave es el desarrollo de modelos “livianos” optimizados para entornos con limitaciones computacionales, manteniendo alta precisión diagnóstica. La tesis también analiza desafíos éticos, como el sesgo en conjuntos de datos predominantemente caucásicos, proponiendo técnicas de aumento de datos para mejorar equidad en diagnósticos para minorías étnicas.

Según Sebastián, (2022), el estudio se enfoca en salud fetal mediante IA explicable (*XAI*). Combina modelos predictivos con interfaces interpretables que permiten a obstetras entender las razones detrás de cada recomendación (ej: riesgo de preeclampsia o restricción de crecimiento intrauterino). La metodología compara técnicas como *SHAP* (*Shapley Additive Explanations*) y *LIME* (*Local Interpretable Model-agnostic Explanations*) aplicadas a datos de monitoreo fetal, historiales maternos y ecografías. Los resultados muestran que estos sistemas no solo mantienen alta precisión ( $\geq 90\%$  en predicción de *distress* fetal), sino que aumentan la confianza clínica al mostrar factores contribuyentes en formato comprensible. La implementación piloto en hospitales ecuatorianos redujo falsos positivos en un 15 % respecto a métodos tradicionales.

### 2.3.3. Antecedentes Artículo Científico

Según Nelly, (2024), la investigación desarrolla un sistema inteligente para optimizar la fijación de precios en microempresas del sector hortícola. Emplea técnicas de machine learning y análisis predictivo que consideran variables como costos de producción, fluctuaciones de mercado, demanda estacional y competencia local. El sistema utiliza algoritmos de series temporales (ARIMA) y regresión múltiple para generar recomendaciones dinámicas de precios que maximizan la rentabilidad manteniendo la competitividad. La validación en mercados locales mostró una mejora del 15-20 % en márgenes de utilidad, evidenciando su potencial para apoyar la sostenibilidad de pequeños agricultores. Además, aborda desafíos como la volatilidad en precios de productos perecederos e integración con datos de mercado en tiempo real.

Según Marisela, (2022), el trabajo propone un sistema recomendador basado en algoritmos de clustering no supervisado (K-means, DBSCAN) para agrupar pacientes con características clínicas similares y generar recomendaciones terapéuticas personalizadas. Analiza historiales médicos con diagnósticos, alergias, respuestas previas a tratamientos y comorbilidades. Los clusters identificados permiten sugerir protocolos con un 92 % de precisión en pruebas en hospitales ecuatorianos. Un aporte clave es un módulo de retroalimentación que ajusta recomendaciones según resultados posteriores, creando un ciclo de mejora continua. La implementación redujo errores de medicación en un 30 % y optimizó tiempos de prescripción.

Según Manuel, (2025), la investigación desarrolla un sistema experto con lógica difusa para crear planes nutricionales personalizados en estudiantes de secundaria. El sistema integra datos antropométricos, análisis sanguíneos, nivel de actividad física y preferencias culturales, utilizando un motor de reglas difusas para manejar incertidumbre en datos adolescentes y generar menús equilibrados. Validado en colegios de Lambayeque, mostró un 18 % de mejora en adherencia nutricional, 12 % de reducción en obesidad infantil y mejoras en rendimiento académico. Incluye un componente educativo con gamificación para promover hábitos saludables.

Según Ramiro, (2022), el estudio aplica Support Vector Machines (SVM) para desarrollar el pensamiento computacional en educación primaria. Analiza desempeño estudiantil en actividades lógico-matemáticas considerando patrones de error, tiempo de resolución y estrategias cognitivas. Clasifica a estudiantes en tres niveles y recomienda ejercicios adaptados. La implementación en aulas peruanas mostró 22 % de mejora en resolución de problemas, 15 % de aumento en creatividad algorítmica y reducción de la brecha de género en STEM. Además, incluye un dashboard para docentes con analíticas predictivas.

Según Augusto, (2016), la investigación propone un modelo híbrido (filtrado colaborativo + basado en contenido) para sistemas de recomendación educativa aplicado a un curso de Programación Web. Combina historial académico, estilos de aprendizaje (VARK) y metadata de recursos. Logró un 85 % de precisión en recomendaciones de materiales y actividades prácticas. Aunque desarrollado antes del auge del deep learning, sentó bases metodológicas para sistemas adaptativos en educación superior, con un enfoque en reducir la deserción mediante personalización.

## **2.4. Justificación del Proyecto de Innovación / Mejora / Creatividad**

### **2.4.1. Justificación Teórica**

El presente proyecto se fundamenta en teorías de inteligencia artificial aplicada a la salud, procesamiento de lenguaje natural (PLN) y el desarrollo de tecnologías móviles. Estas disciplinas han demostrado ser efectivas en la mejora del acompañamiento clínico, la asistencia remota y la toma de decisiones informadas por parte del paciente.

Según Bendezu Castilla y Ysla Parra, (2020), el uso de aplicaciones móviles con inteli-

gencia artificial para brindar recomendaciones personalizadas puede influir positivamente en el tratamiento de condiciones clínicas específicas, al proporcionar orientación inmediata a los usuarios basada en datos contextuales. Aunque en su caso se orienta a la nutrición infantil, el enfoque de sistemas de recomendación es extrapolable a otros campos de la salud, como la rehabilitación física.

En el caso de la Clínica RENE Rehabilitación S.A.C., el sistema propuesto busca complementar el tratamiento presencial con una herramienta digital que oriente al paciente durante su recuperación en casa. La integración de una API de inteligencia artificial (Gemini) permite generar respuestas personalizadas ante consultas frecuentes sobre síntomas menores, lo que se alinea con el paradigma actual de atención médica asistida por tecnología.

Así, esta propuesta no solo responde a un vacío operativo, sino que también se sustenta en marcos teóricos vigentes que validan el uso de la IA como herramienta de apoyo en contextos clínicos y terapéuticos.

#### **2.4.2. Justificación Personal**

Como estudiante de Ingeniería de Software con orientación en inteligencia artificial y soluciones móviles, he identificado en esta iniciativa una oportunidad concreta para aplicar mis conocimientos técnicos en un entorno real, con un propósito humano trascendente.

La posibilidad de contribuir a la mejora de la calidad de vida de pacientes que atraviesan procesos de rehabilitación neuromotora me motiva profundamente, no solo desde una perspectiva académica, sino también desde una vocación de servicio. Este proyecto representa para nosotros un puente entre la tecnología y la empatía, al desarrollar una herramienta que facilite el acceso a recomendaciones terapéuticas, optimice la gestión de citas médicas y alivie la carga operativa del personal clínico.

Asumir este reto ha fortalecido nuestro perfil profesional, reafirmando mi interés por construir soluciones tecnológicas que no solo resuelvan problemas técnicos, sino que respondan a necesidades humanas reales.

#### **2.4.3. Justificación Práctica**

La atención post-clínica en tratamientos de rehabilitación neuromotora suele presentar dificultades como el seguimiento poco preciso, el incumplimiento de ejercicios domiciliarios y la falta de contacto directo con el terapeuta. Este proyecto propone una solución práctica a través de una aplicación móvil que permite consultar recomendaciones inteligentes mediante la API de Gemini y recibir recordatorios terapéuticos, mientras que desde el panel administrativo se gestionan las citas y se asignan terapeutas de manera eficaz, optimizando la operatividad del centro médico.

#### **2.4.4. Justificación Tecnológica**

La propuesta de implementar un aplicativo móvil para una clínica de terapia física, potenciado con un Sistema de Recomendación de Salud basado en Inteligencia Artificial (IA), responde a la necesidad de modernizar la atención terapéutica, automatizar los procesos de gestión clínica y ofrecer orientación personalizada y oportuna a los pacientes.

Esta solución tecnológica se alinea con las tendencias actuales de transformación digital en el ámbito de la salud, mejorando tanto la experiencia del usuario como la eficiencia operativa del centro médico. A continuación, se detallan los principales fundamentos tecnológicos que justifican la viabilidad e innovación del proyecto:

- **Atención Inteligente Personalizada** La incorporación de una API de Gemini (IA generativa) permite ofrecer recomendaciones de salud terapéutica personalizadas, según los síntomas o condiciones físicas del paciente. Esto transforma la consulta tradicional en una experiencia proactiva, accesible y disponible en cualquier momento, reduciendo la dependencia exclusiva del horario clínico o de la disponibilidad del personal.
- **Automatización de Procesos Clínicos** La aplicación integra funcionalidades como el registro y login de pacientes, así como un sistema de reserva y gestión de citas automatizado. Esto reduce la carga administrativa del personal, minimiza errores humanos y mejora los tiempos de respuesta al usuario. El panel de administración permite la asignación, modificación o cancelación de citas de manera sencilla y eficiente.
- **Notificaciones Automatizadas vía SMS** Una característica innovadora es la notificación automática vía mensajería SMS, que informa al paciente sobre la hora y el terapeuta asignado. Esta función refuerza la comunicación clínica, evita ausencias por olvido y mejora la puntualidad y compromiso del paciente con su proceso terapéutico.
- **Experiencia de Usuario Mejorada** El diseño de la app se enfoca en la usabilidad, permitiendo a los pacientes navegar intuitivamente entre funciones como reservar citas, revisar recomendaciones personalizadas o comunicarse con el centro. Al ofrecer un entorno centralizado, moderno y fácil de usar, se promueve una mejor adherencia al tratamiento.
- **Integración con Tecnologías de IA de Última Generación** El uso de Gemini como motor de IA garantiza acceso a un sistema de procesamiento natural del lenguaje de última generación, capaz de interpretar preguntas del usuario, comprender contextos clínicos básicos y ofrecer respuestas confiables. Esto coloca a la clínica a la vanguardia de la innovación médica digital.
- **Trazabilidad y Registro de Actividades** Cada interacción dentro del aplicativo queda registrada: reservas realizadas, fechas de consulta, respuestas generadas por la IA, entre otros. Esta información es valiosa para auditorías internas, análisis de satisfacción y seguimiento del comportamiento del paciente.
- **Escalabilidad y Adaptabilidad** El aplicativo está diseñado con una arquitectura escalable que permite incorporar nuevas funcionalidades en el futuro, como video-llamadas con terapeutas, historial clínico digital o integraciones con wearables de salud. Además, puede adaptarse a múltiples sedes o servicios dentro de la misma clínica.
- **Reducción de Costos a Mediano y Largo Plazo** Aunque la implementación inicial de un sistema con IA y gestión automatizada implica una inversión, se traduce en ahorros significativos a futuro: menor gasto en personal administrativo, reducción

de errores manuales, eliminación de llamadas para confirmar citas y disminución de ausencias por falta de recordatorios.

- **Cumplimiento de Normativas de Seguridad y Privacidad** El sistema considera el uso de buenas prácticas en protección de datos personales, cumpliendo con las normativas de privacidad en salud. Se implementan medidas como cifrado de información sensible, autenticación segura y manejo responsable de datos clínicos.
- **Justificación Legal** El sistema se desarrolla en cumplimiento de la legislación peruana sobre protección de datos personales Congreso de la República del Perú, (2021) y las normativas internacionales de privacidad, asegurando que la información médica y biométrica de los pacientes sea tratada con confidencialidad, integridad y seguridad.

## 2.5. Marco Teórico y Conceptual

### 2.5.1. Fundamentos Teóricos del Proyecto de Innovación / Mejora / Creatividad

#### 2.5.2. Bases Teóricas

##### 2.5.2.1. Variable 1: Chat de Recomendaciones

###### Fundamentos Conceptuales del Chat de Recomendaciones

**Definición de Chat con Inteligencia Artificial** Según Pérez Marín, (2020), un chat con inteligencia artificial es una herramienta basada en algoritmos de procesamiento de lenguaje natural (PLN) que permite interactuar con los usuarios a través de conversaciones automatizadas, proporcionando respuestas coherentes, personalizadas y en tiempo real. Estas tecnologías simulan el lenguaje humano y aprenden del comportamiento del usuario, adaptándose progresivamente para ofrecer recomendaciones más precisas a medida que aumenta el volumen de interacción.

**Evolución de los Sistemas Conversacionales** En sus primeras etapas, los sistemas conversacionales se basaban en estructuras rígidas y respuestas predefinidas, conocidos como *chatbots de flujo* o *bots de árbol*, que empleaban árboles de decisión y patrones estáticos. Estos modelos carecían de adaptabilidad contextual y capacidad de razonamiento. Sin embargo, con el surgimiento del aprendizaje profundo y las redes neuronales, especialmente los modelos transformadores, se logró una evolución sustancial en la capacidad de entender intenciones, mantener coherencia contextual e incluso reconocer emociones del usuario Jurafsky y Martin, (2023).

**Avances Tecnológicos: De BERT y GPT a Gemini** Uno de los hitos clave ha sido la aparición de modelos de lenguaje de gran escala (*Large Language Models*, LLMs), como BERT, GPT y recientemente Gemini, desarrollado por (G. DeepMind, (2023)). Gemini se distingue por ser un modelo multimodal con arquitectura Transformer optimizada mediante un sistema *Mixture of Experts* (MoE). Además, ha sido entrenado mediante técnicas de auto-supervisión multimodal y ajustado con aprendizaje supervisado y por refuerzo, permitiéndole comprender texto, imágenes y otros formatos de entrada de forma integrada Google Cloud, (2024).

**Aplicaciones Clínicas de la IA Conversacional** En el ámbito clínico, estos modelos permiten desarrollar asistentes virtuales que no solo responden a preguntas frecuentes, sino que también entregan recomendaciones terapéuticas, validan criterios médicos y hacen seguimiento personalizado. Según Chatterjee et al., (2022), los chatbots médicos han mejorado significativamente la adherencia a tratamientos, la educación del paciente y han reducido la carga operativa del personal clínico.

**Integración Técnica en Android Studio mediante APIs** La integración de Gemini en aplicaciones móviles se logra a través de APIs generativas conectadas mediante claves seguras (API Key) y llamadas HTTP. Esta conexión permite interpretar la entrada textual del paciente, procesarla con inteligencia artificial, y devolver recomendaciones

personalizadas de fisioterapia. La implementación en Android Studio permite mantener un flujo de comunicación eficiente y escalable. Como menciona Google Cloud, (2024), la integración con Gemini posibilita la detección de intención médica, análisis de síntomas mediante lenguaje natural y evaluación empática del paciente.

## Ramas Interrelacionadas y Tecnologías Fundamentales

**Procesamiento del Lenguaje Natural (PLN)** El núcleo del sistema conversacional está en el Procesamiento del Lenguaje Natural, una subdisciplina de la inteligencia artificial que permite que las máquinas comprendan, interpreten y generen lenguaje humano. Gracias a esta tecnología, el chat es capaz de comprender preguntas de los pacientes, detectar síntomas, y ofrecer recomendaciones terapéuticas comprensibles y útiles, en base a parámetros clínicos definidos por especialistas Jurafsky y Martin, (2023).

**Modelos de Aprendizaje Automático** El motor del sistema está basado en modelos de lenguaje avanzados, como Gemini, que utilizan redes neuronales profundas y técnicas de aprendizaje automático para mejorar constantemente sus respuestas. A medida que el sistema interactúa con más usuarios, aprende patrones, necesidades recurrentes y estilos de comunicación, incrementando la calidad y precisión de sus sugerencias terapéuticas Anil et al., (2023); Brown et al., (2020).

**Inferencia basada en Datos Médicos** El sistema incorpora reglas y patrones obtenidos a partir de guías clínicas de fisioterapia, protocolos de atención y bases de datos médicas, que permiten que la IA ofrezca recomendaciones ajustadas a cada perfil de paciente. Esta tecnología se alinea con el paradigma de la medicina personalizada, donde las soluciones se adaptan al contexto clínico individual Topol, (2019).

**Interacción Multicanal y Automatización** A través del uso de una API Key de Gemini, el sistema puede integrarse con otros módulos tecnológicos, como la reserva automática de citas, la notificación vía SMS al paciente, y la actualización en tiempo real del sistema de gestión. Estas integraciones potencian la automatización del flujo de atención, lo cual libera carga operativa al personal administrativo y mejora la experiencia del paciente Google Cloud, (2024).

**Seguridad y Privacidad de la Información** Al manejar información sensible relacionada con la salud, el sistema incorpora medidas de seguridad basadas en el cifrado de datos, autenticación mediante credenciales seguras, y protocolos compatibles con normativas como la Ley de Protección de Datos Personales. Esto garantiza que tanto las recomendaciones como los historiales de interacción sean tratados de forma confidencial y segura International Organization for Standardization, (2021); U.S. Department of Health and Human Services, (2020).

## Análisis Técnico del Chat de Recomendaciones Basado en Gemini

**Métodos y Procesos del Chat de Recomendaciones** El sistema implementado en esta aplicación móvil sigue un enfoque de procesamiento conversacional secuencial basado en el patrón Request-Response, utilizando inteligencia artificial generativa mediante

la API de Gemini 1.5 Flash. Este modelo fue desarrollado por el equipo de Google Deep-Mind, quienes, como señalan en la documentación técnica oficial (Google Cloud, (2024)), han integrado una arquitectura basada en Transformers, aprendizaje por refuerzo y mecanismos avanzados de atención dinámica del tipo Mixture of Experts (MoE). Anakin, (2024) también destaca que esta combinación optimiza la eficiencia de las respuestas, especialmente en entornos clínicos o móviles, donde la velocidad de respuesta y el consumo de recursos son críticos.

## Características del Método

- **Entrada Única:** El sistema acepta como entrada una única cadena de texto natural, como por ejemplo una descripción de síntomas o consultas sobre fisioterapia. Según Jurafsky y Martin, (2023), este tipo de entrada directa es esencial para simplificar el diseño del flujo conversacional y mantener la interpretación contextual del lenguaje humano.
- **Procesamiento Lineal:** La estructura conversacional sigue un flujo secuencial sin bifurcaciones lógicas. Esta característica permite reducir la complejidad computacional y facilita el diseño del flujo de interacción, como también argumenta Jurafsky y Martin, (2023) al referirse a modelos conversacionales simples en sistemas especializados.
- **Salida Única:** El modelo genera una única recomendación estructurada en cada mensaje. Esta estrategia minimiza la ambigüedad y mejora la claridad en aplicaciones médicas, según se explica en la guía técnica de Jurafsky y Martin, (2023) sobre procesamiento del lenguaje en sistemas sensibles.
- **Estado Conversacional:** El sistema conserva el contexto de la conversación para mantener la coherencia entre turnos. Esta capacidad se logra mediante ventanas de contexto y mecanismos de memoria temporal, como describen detalladamente Jurafsky y Martin, (2023) en su análisis sobre gestión del estado conversacional en modelos de lenguaje.

**Procesos Técnicos del Sistema Conversacional** El diseño funcional del chat fue dividido en cuatro etapas técnicas principales, cada una de las cuales tiene un rol específico en el flujo de procesamiento:

### a. Proceso de Entrada (Input Processing)

- Se inicia con la captura de texto proporcionado por el usuario.
- Luego se ejecuta una validación sintáctica básica del mensaje.
- Se procede con la sanitización de los datos para eliminar posibles caracteres maliciosos.
- Finalmente, se contextualiza el mensaje para integrarlo correctamente en el historial de conversación, asegurando coherencia, tal como lo sugieren Jurafsky y Martin, (2023) en sus recomendaciones para la robustez conversacional.

### b. Proceso de Análisis (Analysis Processing)

- Se realiza la tokenización y análisis gramatical de la estructura del mensaje.
- A continuación, se efectúa un análisis semántico para interpretar el significado global.
- Finalmente, se extraen entidades médicas clave, como síntomas o tratamientos, utilizando técnicas de reconocimiento de entidades nombradas (NER), descritas por Chiu y Nichols, (2016) como fundamentales para la extracción precisa de información en textos clínicos.

#### c. Proceso de Generación (Generation Processing)

- Primero, se construye un prompt personalizado a partir del análisis anterior.
- Este prompt es enviado a la API de Gemini para su procesamiento.
- La respuesta generada por el modelo es interpretada y adaptada.
- Finalmente, se aplica un post-procesamiento para ajustar el contenido a la necesidad del usuario, en línea con las prácticas descritas por Google Cloud, (2024) en el uso eficiente de Gemini.

#### d. Proceso de Salida (Output Processing)

- La respuesta pasa por filtros de seguridad y validación médica.
- Luego se formatea para su correcta visualización en la interfaz.
- Se renderiza en la UI desarrollada en Android Studio.
- Y finalmente, se registra en el historial de conversación del usuario, lo cual sigue el protocolo de diseño de interfaces conversacionales seguras, tal como destacan Google Cloud, (2024) y Jurafsky y Martin, (2023).

**Función Principal en Android Studio** El método central que estructura la lógica del sistema se define como:

```
procesarConsultaPaciente(entrada: String): Recomendacion
```

**Contextualización del Uso de Gemini en el Proyecto** Es importante aclarar que, en este proyecto, el modelo **Gemini 1.5 Flash** no ha sido entrenado ni ajustado manualmente por los autores de esta tesis. En su lugar, se ha integrado a través de una API externa proporcionada por Google Cloud. La API permite el acceso a un modelo ya entrenado mediante solicitudes HTTP, lo cual facilita su implementación en aplicaciones móviles sin requerir conocimientos avanzados en entrenamiento de modelos de lenguaje.

Los conceptos presentados previamente —como la arquitectura Transformer, el mecanismo de autoatención (Self-Attention), el aprendizaje por refuerzo con retroalimentación humana (RLHF) y el Mixture of Experts (MoE)— han sido incluidos en este marco teórico con el objetivo de comprender las capacidades internas del modelo utilizado. Estos principios permiten justificar por qué Gemini puede generar recomendaciones terapéuticas personalizadas con alta precisión en lenguaje natural, aun cuando el modelo sea consumido como un servicio externo.

Por tanto, el sistema desarrollado en esta investigación se basa en una lógica de *IA como servicio* (*AI-as-a-Service*), donde la inteligencia artificial es invocada remotamente a través de una clave de API y se encarga de procesar entradas textuales para entregar respuestas contextualizadas. Esta estrategia tecnológica ha permitido reducir significativamente el tiempo de desarrollo y enfocarse en la integración clínica y funcional de la solución.

### **Componentes Técnicos Principales**

- Arquitectura *Transformer*, base del procesamiento contextual.
- Sistema *Mixture of Experts* (MoE) para distribuir dinámicamente cargas de atención.
- Entrenamiento auto-supervisado con datos textuales, visuales y multimodales.
- Ajuste fino a través de aprendizaje supervisado y aprendizaje por refuerzo con retroalimentación humana (RLHF).

### **Arquitectura Transformer y su Aplicación en Gemini**

**Introducción a la arquitectura Transformer** Dado que el código y los parámetros exactos del modelo Gemini 1.5 Flash no son de dominio público, se ha recurrido a literatura técnica validada para comprender sus componentes fundamentales. Uno de estos componentes clave es la arquitectura Transformer, propuesta por Vaswani et al. ((2017)) en el artículo *"Attention is All You Need"*, el cual se considera la base conceptual de los modelos de lenguaje de gran escala (LLMs), incluyendo Gemini.

El éxito de esta arquitectura radica en su innovador mecanismo de *self-attention*, mediante el cual cada palabra en una secuencia puede asignar pesos de atención a otras palabras relevantes dentro del mismo contexto. Este mecanismo permite al modelo capturar dependencias semánticas y relaciones lógicas entre términos, lo cual resulta esencial para interpretar correctamente frases complejas del lenguaje natural, como aquellas relacionadas con síntomas o dolencias clínicas.

A diferencia de arquitecturas previas como las redes neuronales recurrentes (RNNs), el Transformer permite un procesamiento paralelo de secuencias, acelerando el entrenamiento y mejorando la comprensión global del texto. Además, incorpora codificaciones posicionales (*positional encodings*) para preservar el orden de las palabras, lo cual es crucial en contextos médicos, donde frases como "no tengo fiebre, pero sí dolor muscular" deben ser interpretadas correctamente en su secuencia lógica.

**Estructura Interna del Transformer** Alammar, (2018), en su recurso didáctico "The Illustrated Transformer", desglosa gráficamente los principales componentes del modelo. A continuación, se explican los elementos clave:

#### **Codificador y Decodificador**

Según (Alammar, (2018)), el Transformer se compone de un codificador que interpreta la entrada textual y un decodificador que genera la respuesta. Esta división permite adaptar el sistema a tareas de comprensión y generación de lenguaje.

#### **Mecanismo de Autoatención (Self-Attention)**

El corazón del Transformer es el mecanismo de autoatención. (Vaswani et al., (2017)) explican que cada palabra de entrada se transforma en vectores de consulta (Q), clave (K) y valor (V), y se pondera su importancia contextual. El cálculo matemático base es:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$

### **Atención Multi-Cabeza**

Esta técnica permite al modelo evaluar múltiples relaciones semánticas de forma simultánea. Según (Vaswani et al., (2017)), las múltiples cabezas de atención mejoran la capacidad de extraer patrones sintácticos y semánticos complejos.

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Las salidas de todas las cabezas se concatenan y se proyectan mediante otra matriz entrenable:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \dots, head_h)W^O$$

### **Embeddings y Positional Encoding**

Según (Vaswani et al., (2017)), las palabras se representan como vectores (embeddings) y se les añade información posicional para mantener el orden secuencial. Esto resulta crucial en tareas clínicas, donde el contexto temporal y espacial de los síntomas es fundamental.

**Aplicación de Transformer en Gemini** Gemini aplica el modelo Transformer para analizar consultas médicas en lenguaje natural. Según (DeepMind, (2024)), esta arquitectura permite interpretar síntomas como "dolor punzante en la parte baja de la espalda" generar recomendaciones clínicas adaptadas.

## **Técnicas de Entrenamiento del Modelo**

**Entrenamiento Auto-supervisado Multimodal** Gemini fue entrenado con grandes cantidades de datos no etiquetados. (G. DeepMind, (2023)) señala que este enfoque permite al modelo aprender correlaciones entre texto, imágenes y otros formatos, esenciales para contextos médicos y fisioterapéuticos.

**Ajuste Fino: Supervisado y por Refuerzo** Ouyang et al., (2022) detallan que el ajuste fino de modelos como Gemini se realiza mediante:

- **Aprendizaje Supervisado:** utilizando ejemplos médicos verificados por humanos.
- **RLHF (Reinforcement Learning from Human Feedback):** en el cual evaluadores humanos califican respuestas, y estas calificaciones mejoran futuras predicciones.

Este proceso incrementa la precisión y seguridad en entornos sensibles como la salud.

**Mixture of Experts (MoE) para Escenarios Clínicos** Gemini 1.5 Flash incorpora MoE como técnica de escalabilidad y especialización. (Zhou et al., (2023)) destacan que esta arquitectura activa solo los expertos necesarios según el tipo de consulta médica, lo cual reduce el uso de recursos y mejora la precisión clínica.

Por ejemplo, ante una consulta que mencione dolor, se activa un submodelo especializado en fisioterapia, mientras que otras entradas podrían activar expertos en medicina general.

### 2.5.2.2. Variable 2: Gestión de Citas Médicas

#### Fundamentos Conceptuales

**Definición** Aplicación móvil desarrollada en *Android Studio* con *Kotlin* para la reserva, gestión y administración de citas médicas. Utiliza *Firebase Firestore* como base de datos en tiempo real y maneja dos roles principales:

- *Pacientes*: Reservan, modifican o cancelan citas.
- *Administradores*: Gestionan terapeutas, asignan horarios y envían notificaciones vía SMS.

**Firebase** LLC, (2024): Plataforma de desarrollo de aplicaciones creada por Google, diseñada para simplificar la creación de proyectos móviles y web. Originalmente lanzada en 2012 y adquirida por Google en 2014, Firebase ofrece herramientas backend como servicio, eliminando la necesidad de gestionar infraestructura compleja.

A diferencia de soluciones tradicionales como MySQL o MongoDB, Firebase adopta un enfoque integrado que combina base de datos en tiempo real, autenticación, hosting y más en un solo producto. Su componente principal, Firestore, es una base de datos NoSQL orientada a documentos que sincroniza datos instantáneamente entre dispositivos, incluso en condiciones de red limitada.

Entre sus características destacadas se encuentran:

- **Firestore**: Almacenamiento flexible con consultas en tiempo real y soporte offline
- **Authentication**: Sistema seguro de acceso con múltiples proveedores (email, Google, Facebook)
- **Cloud Functions**: Ejecución de código backend sin servidores
- **Hosting**: Despliegue rápido con SSL integrado

Firebase es particularmente popular en aplicaciones médicas como sistemas de citas, gracias a su capacidad para manejar datos sensibles con reglas de seguridad configurables y su escalado automático durante picos de demanda. Según estudios recientes, reduce el tiempo de desarrollo de aplicaciones móviles en hasta un 60 % comparado con soluciones tradicionales.

**Objetivo** Optimizar el proceso de reserva, reducir tiempos de espera y garantizar una experiencia de usuario intuitiva mediante:

- Validación automática de datos (DNI, teléfono).
- Notificaciones en tiempo real (SMS o Firebase Cloud Messaging).

**Trabajos Relacionados** En la tesis titulada "Sistema de gestión de citas médicas en Android con Firebase para una clínica en Lima" López y Pérez, (2022), se desarrolló una aplicación con funcionalidades similares que permitía a los usuarios seleccionar un especialista, elegir horarios disponibles y recibir confirmaciones de citas mediante notificaciones push. Este trabajo evidenció que el uso de Firebase como backend permite una integración rápida y segura de funcionalidades críticas como autenticación, sincronización en tiempo real y escalabilidad, lo cual coincide con los objetivos planteados en el presente proyecto.

- En la tesis de Torres y Ríos, (2021) titulada **Aplicación móvil para gestión de citas médicas utilizando** Firebase y Android Studio para centros médicos del Callao", se diseñó una app enfocada en pacientes que reservaban citas de forma remota. Se priorizó el uso de Firebase Authentication y Firestore para permitir autenticación segura y almacenamiento de datos médicos en tiempo real.
- En el artículo de García et al., (2020) publicado en la **Revista Peruana de Ingeniería de Software**, se presentó una arquitectura híbrida para apps médicas con capacidad de agendamiento, gestión de usuarios y envío de alertas por SMS usando Firebase Cloud Messaging y 'SmsManager' nativo de Android .
- Según el estudio de Méndez y Fernández, (2022), el uso de Firestore en entornos médicos garantiza sincronización inmediata de datos críticos como horarios, disponibilidad y confirmaciones de citas, mejorando la experiencia del paciente y reduciendo errores administrativos en un 45 % .
- La tesis de Vásquez y León, (2021), **Sistema móvil de gestión de pacientes para clínicas privadas con base en Firebase y Kotlin**, demostró que la adopción de reglas de seguridad personalizadas en Firestore puede proteger los historiales médicos y segmentar el acceso según los roles definidos (médico, paciente, asistente).
- En un análisis comparativo de plataformas backend realizado por Romero et al., (2023), se concluyó que Firebase supera a otros BaaS como AWS Amplify o Supabase en proyectos móviles pequeños y medianos, especialmente en rapidez de desarrollo, seguridad en tiempo real y facilidad de integración con librerías Android

**Fundamentos Técnicos Adicionales** En el desarrollo de sistemas de citas médicas, algunos principios y tecnologías recurrentes incluyen:

- **Validación de campos médicos:** Implementar funciones que garanticen que el DNI, teléfono o nombre ingresado sea válido reduce significativamente los errores durante el registro y reserva Martínez, (2020).
- **Notificaciones de doble vía:** Combinando 'SmsManager' con Firebase Cloud Messaging se puede cubrir tanto usuarios sin conexión a Internet como aquellos con acceso a redes móviles, garantizando la entrega del recordatorio de cita Chávez, (2022).
- **Prevención de reservas simultáneas:** El uso de 'runTransaction' en Firestore evita colisiones al reservar citas y asegura que un horario no se asigne dos veces G. Developers, (2023).

- **Control de estados de cita:** La mayoría de los sistemas similares implementan estados como "pendiente", "confirmada", "cancelada", "finalizada" para el ciclo completo de cada cita médica, lo cual permite trazabilidad en el historial de atención médica Rodríguez, (2021).

### **2.5.3. Tecnologías y Herramientas**

#### **2.5.3.1. Entorno (Móvil)**

Para implementar esta arquitectura, se utiliza Google Android Studio Team, (2025) Android Studio, el entorno oficial de desarrollo proporcionado por Google. Esta herramienta integrada (IDE) permite a los desarrolladores crear, depurar y probar aplicaciones móviles con una amplia gama de herramientas como el editor de diseño visual, emuladores, análisis de rendimiento y soporte directo para Kotlin y Java. Android Studio facilita la gestión de proyectos, recursos y dependencias, convirtiéndose en el estándar de facto para el desarrollo de apps en Android.

#### **2.5.3.2. Tecnología de Desarrollo Backend**

En el presente proyecto, el backend está implementado directamente en el entorno Android Studio, utilizando el lenguaje **Kotlin** para gestionar la lógica de negocio de la aplicación. Aunque no se cuenta con un servidor tradicional externo, el proyecto incorpora una arquitectura backend centrada en dos ejes fundamentales:

**Lógica interna en Kotlin (in-app backend)** La lógica de negocio (como validación de datos, reglas de reserva de citas, integración de IA) se desarrolla dentro de clases específicas, que se encargan de gestionar el estado de la aplicación, comunicarse con servicios externos y manipular la interfaz según el flujo de datos, siguiendo las recomendaciones de arquitectura de componentes propuestas por A. Developers ((2023)).

**Servicios Cloud vía Firebase y Gemini API** En lugar de un backend tradicional con servidor propio, se emplean servicios de backend como servicio (BaaS) proporcionados por Firebase y Gemini. Según LLC ((2024)), Firebase permite almacenar y sincronizar datos en tiempo real, mientras que la API de Gemini permite generar contenido inteligente desde la nube, como indican DeepMind ((2024)):

**Integración de librerías** El backend se apoya en librerías asíncronas como **Kotlin Coroutines** para ejecutar operaciones en segundo plano, garantizando que el acceso a la base de datos y las respuestas del modelo de IA no bloqueen la interfaz de usuario, tal como recomienda JetBrains ((2024)).

#### **2.5.3.3. Tecnología de Desarrollo Frontend**

En el desarrollo de aplicaciones Android, la estructura fundamental de la interfaz de usuario se organiza mediante Android UI Team, (2025) actividades (Activity) y diseños (Layout). Una Activity representa una pantalla con la que el usuario puede interactuar, mientras que los Layout definidos en XML especifican la disposición visual de los elementos en dicha pantalla, como botones, campos de texto e imágenes. Esta separación entre lógica y presentación permite un diseño modular, limpio y mantenable.

## Librerías Utilizadas

**AndroidX Core KTX** Según Android Developers A. Developers, (2025), Core KTX proporciona extensiones idiomáticas de Kotlin para la biblioteca AndroidX Core, permitiendo escribir código más conciso, expresivo y fácil de mantener en aplicaciones Android.

**Firebase Auth KTX** Firebase, (2025), indica que esta biblioteca facilita la implementación de autenticación de usuarios en aplicaciones móviles, incluyendo login por correo electrónico, número de teléfono, y proveedores externos como Google o Facebook.

**Firebase Firestore KTX** Según Firebase, (2025), esta biblioteca permite almacenar, sincronizar y consultar datos estructurados en la nube, proporcionando una base de datos NoSQL en tiempo real optimizada para entornos móviles.

**Play Services Auth API Phone** De acuerdo con Google, (2025), esta biblioteca permite la verificación automática de números telefónicos mediante la API SMS Retriever, reduciendo la fricción en el proceso de autenticación por SMS.

**AndroidX Lifecycle ViewModel Compose** Android Developers, (2025), destacan que esta biblioteca permite integrar componentes ViewModel con Jetpack Compose, facilitando la separación de lógica de negocio y presentación dentro del ciclo de vida de la UI.

**AndroidX Activity Compose** Según Android Developers, (2025), esta biblioteca permite gestionar el ciclo de vida de actividades y su interacción con Jetpack Compose, modernizando la arquitectura de la interfaz en aplicaciones Android nativas.

### 2.5.3.4. Herramienta de la Base de datos

LLC, (2024) Firebase, una plataforma de desarrollo de aplicaciones móviles de Google que ofrece una solución integral basada en la nube, permitiendo almacenamiento en tiempo real, autenticación y sincronización de datos de forma eficiente. Esta combinación de tecnologías está orientada a ofrecer una experiencia ágil, escalable y segura dentro del ecosistema móvil.

# Capítulo 3

## Análisis de la Situación Actual

### 3.1. Diagrama del Proceso, Mapa del Flujo de Valor y/o Diagrama de Operación Actual

En este Diagrama el proceso inicia cuando el cliente solicita una cita. El sistema verifica la disponibilidad y, si no hay disponibilidad, informa al cliente y le sugiere otra fecha, finalizando así el proceso. Si hay disponibilidad, el sistema procede a registrar los datos del cliente, confirmar la cita y enviar un recordatorio. El día acordado, el cliente llega a la cita, lo que es reconocido tanto por el sistema como por el personal de recepción. Finalmente, en recepción se realiza el registro y la facturación del cliente.

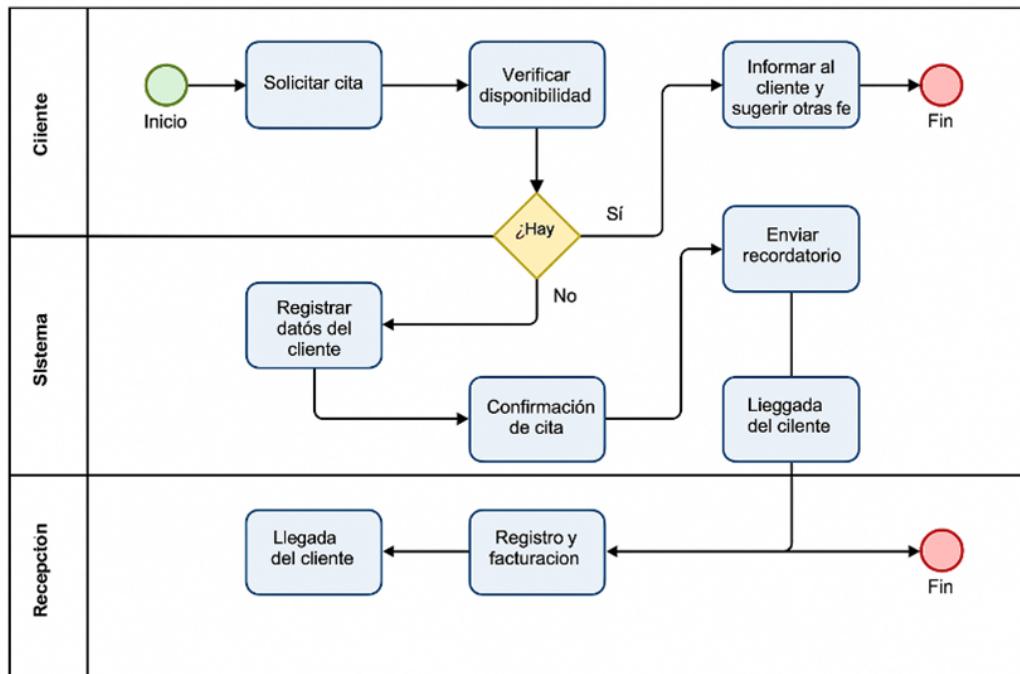


Figura 3.1: Organigrama de la Empresa

### **3.2. Efectos del Problema en el Área de Trabajo o en los Resultados de la Empresa**

Los problemas identificados en la lluvia de ideas reflejan diversas limitaciones que afectan la eficacia del seguimiento terapéutico en entornos clínicos y domiciliarios. Cada una de estas causas tiene efectos concretos en la operación, el uso de recursos y la calidad del servicio prestado.

Cuadro 3.1: *Lluvia de ideas de las causas*

Nº	Lluvia de ideas por fenómenos
CA1	Falta de herramientas automatizadas para seguimiento remoto.
CA2	Recomendaciones terapéuticas generales, no personalizadas.
CA3	Dependencia de datos subjetivos.
CA4	Falta de alertas automáticas sobre retrocesos.
CA5	Baja adherencia a ejercicios en casa.

### **3.3. Análisis de las Causas Raíz que Generan el Problema/Necesidad**

La falta de un sistema centralizado de recomendaciones personalizadas en RENE Reeducación Neuromotora S.A.C. representa un desafío significativo en el seguimiento y tratamiento de los pacientes fuera del centro de rehabilitación. Actualmente, las recomendaciones se basan en observaciones subjetivas y no personalizadas de los terapeutas, lo que impide ajustar el tratamiento según la evolución real de cada paciente. Esta situación genera brechas en el seguimiento, afectando la calidad de la rehabilitación y aumentando el riesgo de abandono del tratamiento. La implementación de un sistema de recomendación de estado de salud con inteligencia artificial es crucial para proporcionar un seguimiento continuo y en tiempo real, adaptando las recomendaciones según las observaciones del terapeuta y los datos objetivos, lo que permitirá una mejor adherencia al tratamiento y una recuperación más efectiva.

## Diagrama de Ishikawa

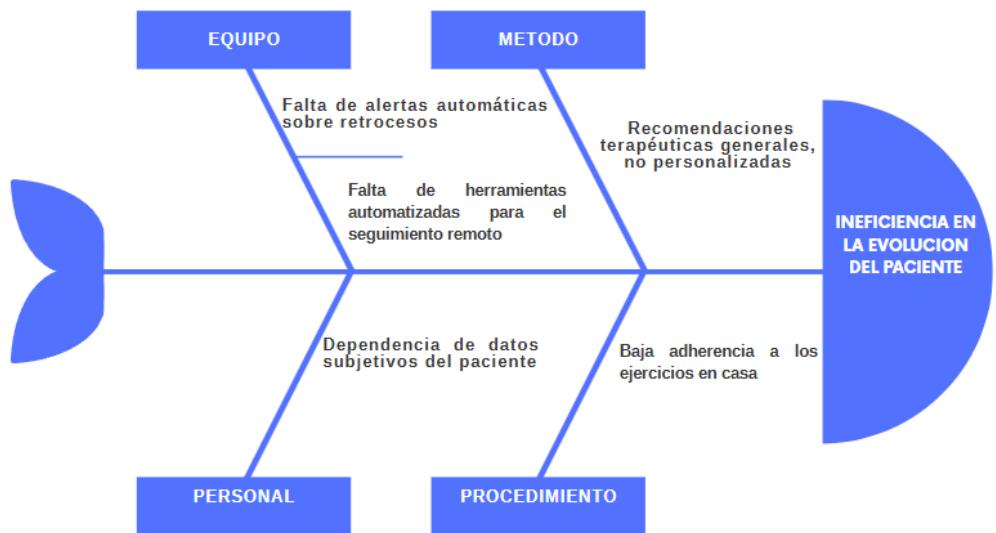


Figura 3.2: Diagrama de Ishikawa

## 3.4. Priorización de Causas Raíz

### 3.4.1. Lista de Causas Encuestadas

Cuadro 3.2: Identificación de Problemas según encuestas

Nº	Identificación del Problema	4 SIEM-PRE	3 CASI SIEM-PRE	2 AL-GU-NAS VE-CES	1 NUN-CA	Total
CA1	¿Cómo calificas la falta de herramientas automáticas para seguimiento remoto?	0	1	3	4	8
CA2	¿Cómo calificas las recomendaciones terapéuticas generales, no personalizadas?	0	2	8	0	10
CA3	¿Cómo calificas la dependencia de datos subjetivos?	0	3	2	1	6
CA4	¿Cómo calificas la falta de alertas automáticas sobre retrocesos?	0	1	4	1	6
CA5	¿Cómo calificas la baja adherencia a ejercicios en casa?	0	1	4	2	7

*Nota:* La tabla presenta los problemas identificados en el sistema actual, evaluados mediante encuestas aplicadas a usuarios y especialistas.

### 3.4.2. Diagrama de Pareto en Función al Grado de la Causa

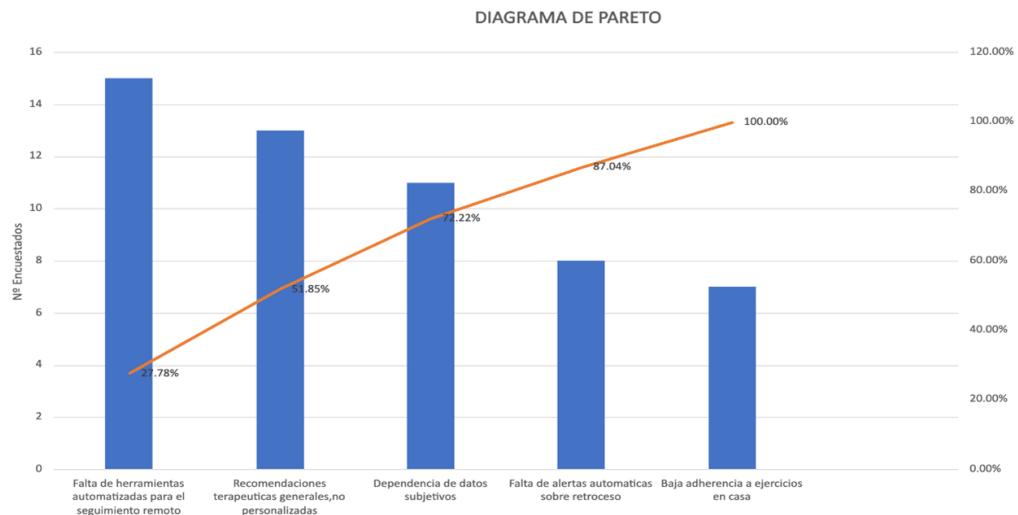


Figura 3.3: Diagrama de Pareto 1

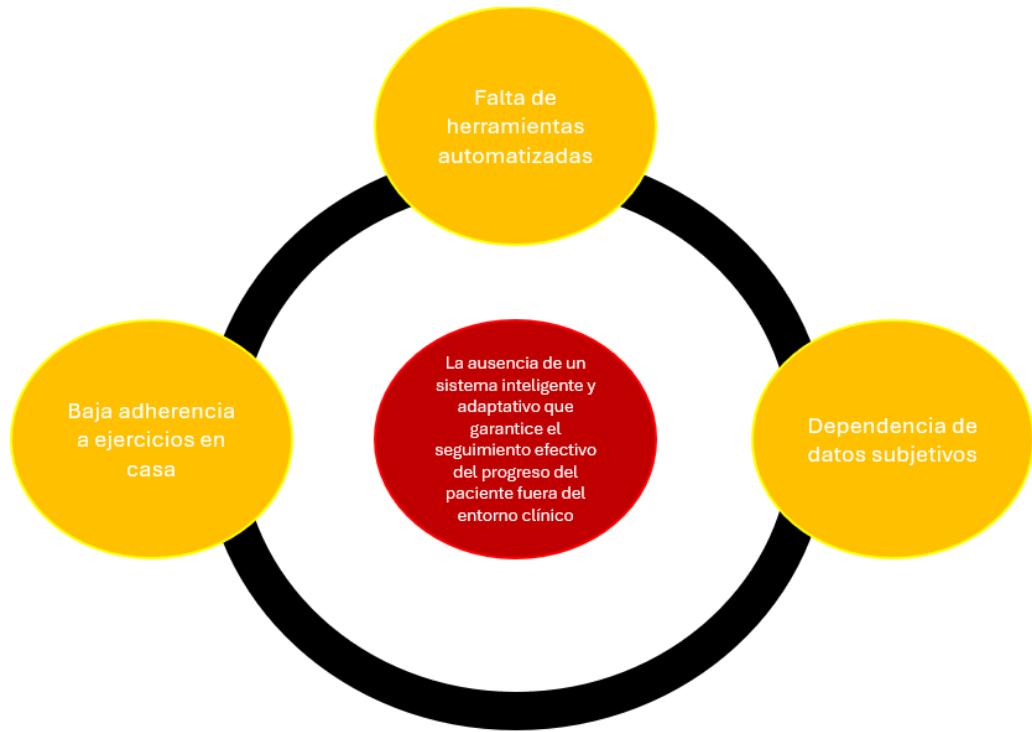


Figura 3.4: Diagrama de Pareto 2

## Análisis Cualitativo y Cuantitativo

### Causa 01: Falta de herramientas automatizadas

La ausencia de herramientas tecnológicas automatizadas impide que el personal clínico monitoree en tiempo real la evolución del paciente fuera del entorno de la clínica. Esto provoca retrasos en la detección de cambios importantes en el estado del paciente,

afectando la capacidad de ajustar los tratamientos a tiempo y reduciendo la efectividad general de la terapia.

#### Análisis cuantitativo:

##### *Impacto de la falta de herramientas automatizadas para seguimiento remoto*

Mes	Pacientes sin seguimiento	Pacientes afectados	Costo por paciente (S/.)	Pérdida mensual (S/.)
Febrero	9	6	50	300
Marzo	10	7	47	330
Abril	11	6	52	310
Mayo	12	8	43	340
Junio	13	8	40	320
Julio	14	7	50	350
Agosto	13	8	43	340
Septiembre	15	9	40	360
Octubre	14	8	41	330
Noviembre	16	10	38	380

*Nota.* Los datos reflejan estimaciones mensuales derivadas de la falta de monitoreo automatizado en pacientes ambulatorios.

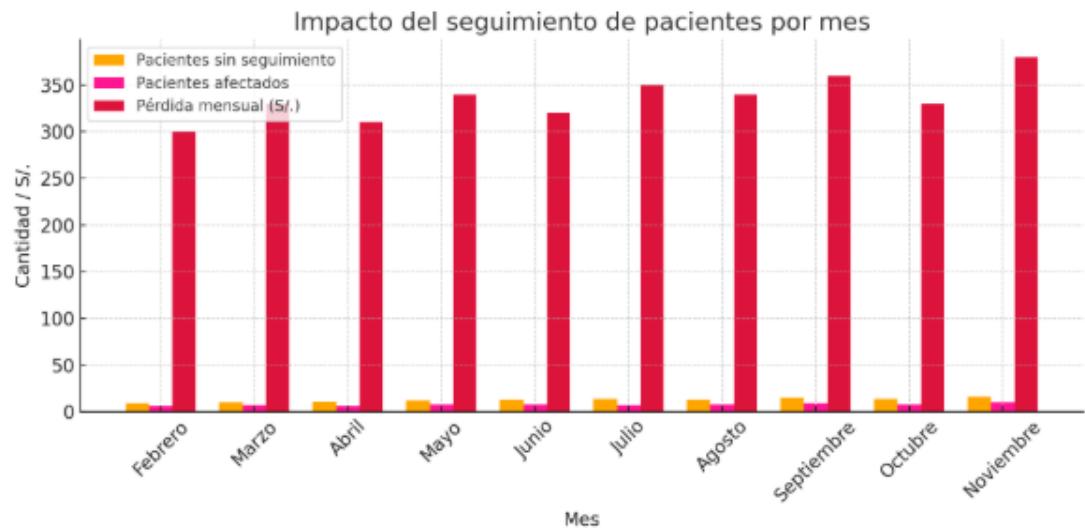


Figura 3.5: Gráfico de la causa I: Falta de herramientas automatizadas

## Causa 02: baja adherencia a ejercicios en casa

Muchos pacientes olvidan realizar los ejercicios asignados en casa, se desmotivan por la falta de seguimiento o abandonan el tratamiento prematuramente. Esta falta de compromiso compromete la continuidad del proceso terapéutico y disminuye los resultados clínicos esperados, prolongando los tiempos de recuperación.

### Análisis cuantitativo:

*Impacto de la baja adherencia a ejercicios en casa*

Mes	Casos estimados	Costo por caso (S/.)	Pérdida mensual (S/.)
Febrero	12	31	370
Marzo	10	40	400
Abril	10	43	430
Mayo	10	41	410
Junio	10	42	420
Julio	10	44	440
Agosto	11	42	460
Septiembre	10	39	390
Octubre	10	44	440
Noviembre	10	44	440

*Nota.* Los datos reflejan pérdidas mensuales estimadas por incumplimiento, desmotivación o abandono de ejercicios domiciliarios por parte del paciente.

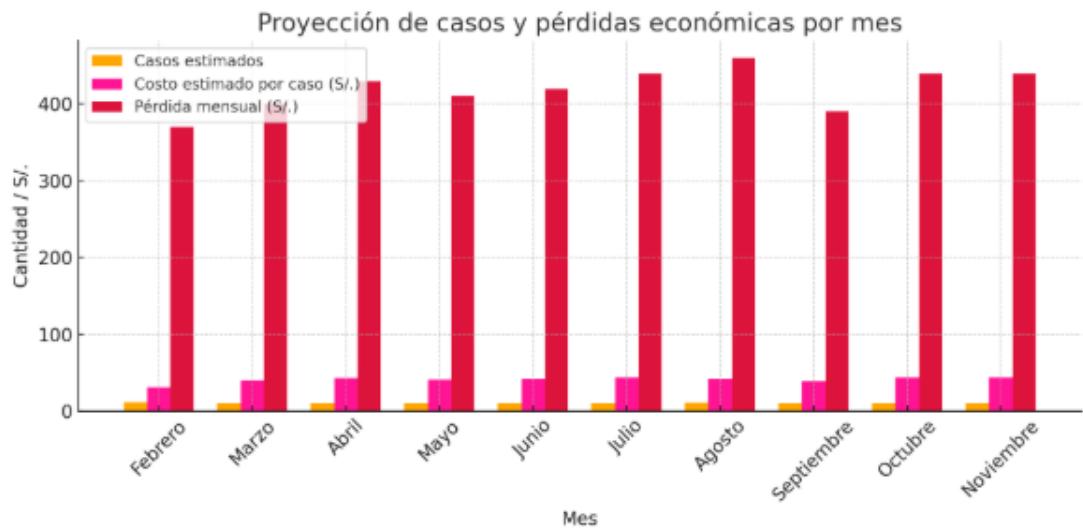


Figura 3.6: Gráfico de la causa II: Recomendaciones no personalizadas

### Causa 03: Dependencia de datos subjetivos

**Análisis cualitativo:** Basarse en información verbal del paciente puede llevar a evaluaciones inexactas, ya que estas percepciones no siempre reflejan el estado real. Esto afecta el diagnóstico y el seguimiento, generando posibles errores en la planificación terapéutica.

#### Análisis cuantitativo:

*Impacto de la dependencia de datos subjetivos en el seguimiento clínico*

Mes	Casos evaluados con datos subjetivos	Errores de diagnóstico	Reajustes tardíos
Febrero	8	4	2
Marzo	9	5	2
Abril	7	3	1
Mayo	10	6	3
Junio	9	4	2
Julio	11	5	3
Agosto	10	4	2
Septiembre	12	6	3
Octubre	11	5	2
Noviembre	12	5	2

*Nota.* La evaluación subjetiva incrementa los errores clínicos y demora los reajustes terapéuticos, afectando el seguimiento del paciente.

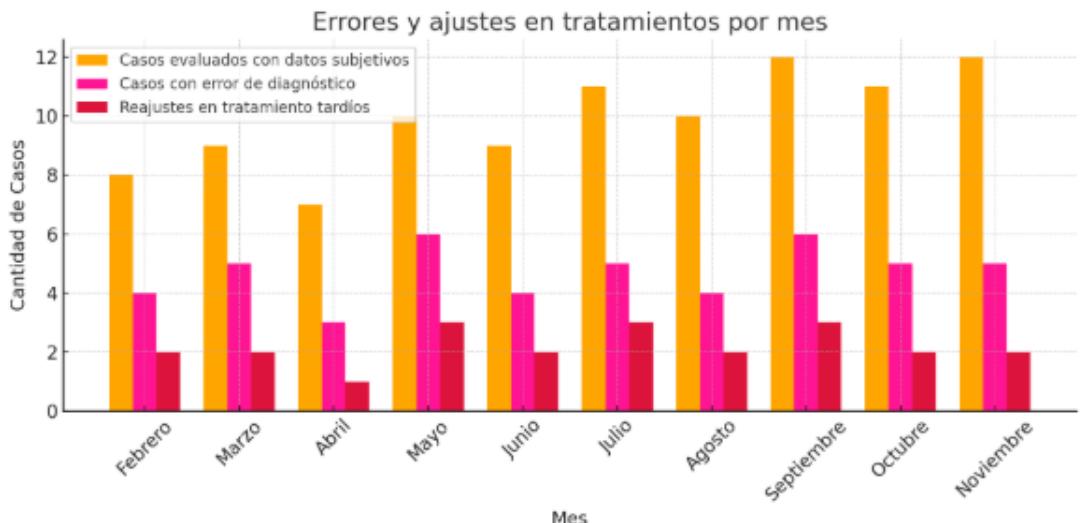


Figura 3.7: Gráfico estadístico de la causa III: Dependencia de datos subjetivos

# Capítulo 4

## Propuesta Técnica de la Mejora

En este capítulo se detallará la implementación de la solución propuesta: una aplicación móvil para la gestión de citas médicas y recomendaciones de salud basada en inteligencia artificial, desarrollada para la Clínica René Reeducación Neuromotora S.A.C.. Para el desarrollo del sistema, se optó por aplicar la metodología ágil Scrumban, la cual combina elementos de Scrum y Kanban. Esta metodología permite una planificación estructurada con entregas iterativas, al mismo tiempo que proporciona la flexibilidad necesaria para adaptarse a los cambios durante el desarrollo.

### 4.1. Plan de acción de mejora Propuesta

Para el desarrollo del presente sistema implementado en la Clínica René, se utilizará la metodología ágil **Kanban**, que permite gestionar el avance de tareas de manera visual y flexible. Esta metodología es ideal para proyectos con requerimientos que pueden evolucionar, ya que facilita la identificación de cuellos de botella, el monitoreo constante del progreso y una colaboración más eficiente entre desarrolladores y personal médico.

La propuesta técnica plantea una mejora integral del proceso de atención médica mediante la incorporación de tecnologías modernas como:

Un módulo de **autenticación por correo y contraseña**, para el ingreso seguro de pacientes y personal.

Un **panel del paciente**, que permite el **registro, reserva de citas** y el acceso a un módulo de **consulta automatizada mediante inteligencia artificial (IA)**, donde se brindan recomendaciones orientativas en salud terapéutica basadas en los síntomas ingresados.

Un **panel administrativo**, desde donde el personal autorizado gestiona manualmente las citas, asigna terapeutas, elimina reservas y automatiza el envío de notificaciones por **SMS** al paciente indicando la hora y el profesional asignado.

La integración con **Firebase**, como plataforma backend para el almacenamiento en tiempo real de usuarios, citas y mensajes, lo que garantiza una sincronización eficiente entre los módulos del sistema.

La incorporación de un modelo de **IA conversacional a través de Gemini API**, que utiliza lógica de procesamiento de lenguaje natural (PLN) para ofrecer respuestas

claras y breves. Esto brinda una primera orientación al paciente sin sustituir al diagnóstico clínico profesional.

Esta mejora responde a la necesidad de optimizar el tiempo del personal médico y empoderar al paciente mediante un canal accesible y automatizado de orientación previa a la atención presencial. Según Yamao y Ernesto, (2020), la incorporación de tecnologías basadas en inteligencia artificial en sistemas de salud promueve un mejor aprovechamiento de los recursos institucionales y mejora la experiencia del usuario al brindar atención preliminar sin depender totalmente de la disponibilidad médica tradicional Yamao y Ernesto, (2020).

Con esta propuesta, el proyecto no solo busca **digitalizar** procesos, sino también **transformarlos** mediante herramientas que favorezcan la eficiencia operativa y la accesibilidad al servicio de salud.

**Entregable: Acta de constitución del Proyecto**

En la actividad presentada se va a desarrollar el Acta de Constitución del Proyecto, en esta se detalla los alcances del proyecto, la duración de este. El acta se encuentra en el Anexo

#### **4.1.1. Fase 1: Iniciación**

**Actividad 1: Creación de la visión del proyecto**

**Descripción:**

Definir de forma clara y comprensible el objetivo general del proyecto, su alcance, metas, beneficios y justificación. Se identificará el problema a resolver y cómo la inteligencia artificial contribuirá a la optimización de los procesos actuales de la Clínica RENE.

Cuadro 4.1: Tabla de visión del proyecto

<b>Visión del Proyecto</b>	Desarrollar una aplicación móvil para la gestión eficiente de citas médicas y el seguimiento del estado de salud de los pacientes fuera del entorno clínico, integrando inteligencia artificial (API Gemini) para brindar recomendaciones básicas de salud en tiempo real.
<b>Sector Objetivo</b>	Salud / Rehabilitación neuromotora
<b>Nombre del responsable – Área</b>	Miguel Ángel Cerna
<b>Necesidades</b>	Optimizar la gestión de citas médicas, reducir la carga operativa del personal, proporcionar orientación médica básica personalizada entre sesiones y mejorar la comunicación entre pacientes y la clínica.
<b>Producto</b>	Aplicación móvil que permita registrar, modificar y cancelar citas, integrar recomendaciones de salud mediante IA, enviar recordatorios automáticos y mejorar la experiencia del paciente.
<b>Valor</b>	<ul style="list-style-type: none"> <li>▪ Mejora en la atención y continuidad del tratamiento.</li> <li>▪ Reducción de errores humanos en la programación de citas.</li> <li>▪ Aumento de la autonomía del paciente en su proceso de recuperación.</li> <li>▪ Modernización de la clínica como institución innovadora.</li> </ul>
<b>Visión planteada</b>	Ser una herramienta digital confiable e inteligente que fortalezca el vínculo entre el paciente y la clínica, facilitando un seguimiento personalizado, automatizado y eficiente de la rehabilitación.
<b>Requerimiento del Software</b>	<ul style="list-style-type: none"> <li>▪ Aplicación desarrollada en Android Studio con Kotlin.</li> <li>▪ Integración con Firebase para autenticación y base de datos.</li> <li>▪ Comunicación con la API de Gemini para recomendaciones de salud.</li> <li>▪ Notificaciones push para recordatorios y confirmaciones de citas.</li> </ul>
<b>Requerimiento del Hardware</b>	<ul style="list-style-type: none"> <li>▪ Dispositivos móviles con sistema operativo Android.</li> <li>▪ Conectividad a Internet.</li> <li>▪ Servidores que permitan el consumo de API externa (Gemini) y el manejo de Firebase en tiempo real.</li> </ul>

### **Entregable: Acta de constitución del Proyecto**

En la actividad presentada se va a desarrollar el Acta de Constitución del Proyecto, en esta se detalla los alcances del proyecto, la duración de este. El acta se encuentra en el Anexo

### **Actividad 2: Identificación de los Stakeholders y Product Owner**

#### **Descripción:**

En esta actividad se identifican los actores clave del proyecto, incluyendo a los **Stakeholders** y al **Product Owner**, quienes representan los intereses de los usuarios y definen las prioridades del producto.

Cuadro 4.2: *Identificación de cargos y responsables*

Código	Nombre	Cargo	Horas
PO1	Miguel Cerna	Product Owner	1 h
ST1	Miguel Cerna	Stakeholder	1 h

Cuadro 4.3: *Equipo de desarrollo del proyecto*

Código	Nombre	Cargo	Horas
SC1	Kevin	Desarrollador	805 h
SC2	Sebastian	Desarrollador	800 h
SC3	Fabian	Desarrollador	800 h

### **Actividad 3: Establecimiento del equipo de desarrollo SCRUM**

En la actividad se va a describir y recopilar las necesidades que va a pedir el Stakeholders para los requerimientos de la construcción del aplicativo o proyecto.

Cuadro 4.4: *Requerimientos funcionales del sistema*

<b>CODREQ</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Rol</b>
RF-001	Iniciar sesión	Permitir a los usuarios iniciar sesión en la aplicación usando credenciales.	Paciente, Administrador
RF-002	Registro de usuario	Proporcionar un formulario para que nuevos usuarios se registren en la app.	Paciente
RF-003	Panel de citas registradas	Mostrar todas las citas que el usuario ha programado, incluyendo detalles.	Paciente, Administrador
RF-004	Creación de cita	Permitir a los usuarios agendar nuevas citas con terapeutas disponibles.	Paciente
RF-005	Cancelación de cita	Opción para que los pacientes cancelen citas previamente programadas.	Paciente
RF-006	Notificaciones de recordatorio	Enviar notificaciones a los pacientes sobre sus citas programadas.	Paciente
RF-007	Gestión de terapeutas	Permitir a los administradores añadir, modificar o eliminar información de terapeutas.	Administrador
RF-008	Asignación de terapeutas	Asignar un terapeuta a un paciente basado en sus necesidades.	Administrador
RF-009	Historial de citas	Permitir a los pacientes ver su historial de citas pasadas y tratamientos.	Paciente
RF-010	Administración de datos del paciente	Permitir a los administradores gestionar los datos de los pacientes.	Administrador
RF-011	Visualización de cronograma de horario	Mostrar a los usuarios los horarios disponibles de los terapeutas.	Paciente, Administrador

Cuadro 4.5: *Requerimientos no funcionales del sistema*

<b>CODREQ</b>	<b>Nombre</b>	<b>Descripción</b>
RNF-001	Rendimiento	La aplicación debe cargar cada página en menos de 2 segundos en una conexión normal.
RNF-002	Usabilidad	La interfaz debe ser intuitiva y fácil de navegar para todos los usuarios.
RNF-003	Seguridad de Datos	Asegurar que toda la información del usuario esté encriptada y sea segura.
RNF-004	Compatibilidad	La aplicación debe ser compatible con las versiones más recientes de iOS y Android.
RNF-005	Escalabilidad	La aplicación debe poder soportar un creciente número de usuarios sin degradar su rendimiento.
RNF-006	Mantenimiento	El sistema debe garantizar actualizaciones regulares sin interrupciones de servicio.
RNF-007	Accesibilidad	Asegurar que la aplicación sea accesible para personas con discapacidades (por ejemplo, soporte para lectores de pantalla).
RNF-008	Disponibilidad	La aplicación debe tener una disponibilidad del 99.5 % a lo largo del mes.
RNF-009	Integración	La aplicación debe poder integrarse con sistemas existentes, como bases de datos de pacientes.
RNF-010	Cronograma de Horario	Facilitar la configuración de horarios para citas de manera sencilla y eficiente por parte de los administradores.

#### 4.1.2. Fase 2: Planificación

Actividad 1: Revisión de Producto Backlog

En cada interacción planificada, se incluirán todas las actividades relacionadas con la construcción y entrega del producto. Esto permitirá a la empresa revisar y verificar si los requisitos se alinean con sus expectativas.

Cuadro 4.6: Tabla del Product Backlog

<b>CODPRO</b>	<b>NOMBRE DE ACTIVIDAD</b>
PB1	Definición del Objetivo del proyecto
PB2	Realizar especificación de funcionalidad
PB3	Recopilar requisitos
PB4	Análisis de requerimiento
PB5	Matriz de Requerimiento
PB6	Caso de Uso
PB7	Desarrollo BD Firebase
PB8	Implementar plantilla de app móvil
PB9	Creación de login y conexión BD Firebase
PB10	Creación de registro y conexión BD Firebase
PB11	CRUD Citas, Categoría en Kotlin
PB12	Elaboración de cronograma de horarios
PB13	Creación de citas
PB14	Cancelación de citas
PB15	Visualización de historial de citas
PB16	Creación de reportes cotización en PDF
PB17	CRUD de prueba
PB18	Pruebas de inserción de citas según su proveedor BD Firebase
PB19	Prueba de validar registros de usuarios

Cuadro 4.7: *Matriz de impacto de prioridades*

<b>CODPRI</b>	<b>Nombre de prioridades</b>	<b>Nivel</b>
M1	Alta	1
M2	Media	2
M3	Baja	3

Cuadro 4.8: *Etiquetas para la clasificación del desarrollo del proyecto*

<b>CODETI</b>	<b>Nombre de etiqueta</b>
ET1	Inicio
ET2	Planificación e investigación
ET3	Diseño e implementación
ET4	Creación de reportes
ET5	Pruebas de CRUD y formularios

## Actividad 2: Integracion del Matriz de Impacto, Prioridades y Mapping:

En esta sesion se detallaran las matrices de impacto de prioridades y de etiquetas, con el fin de proporcionar una descripcion clara del proyecto que se desarrollara.

Tabla Mapping para la relación de los requerimientos del proyecto (Parte 1)

COD MAP	CODE TI	ETI QUETA	COD PRO	NOMBRE ACTIVIDAD	COD ENC	ENCARGADO	ESTIMACIÓN	COD PRI	PRIORIDAD
MAP1	ET1	Inicio	PB1	Definición del Objetivo del proyecto	SC1	Miguel Cerna	2ph	M1	Alta
MAP2	ET1	Inicio	PB2	Realizar especificación de funcionalidad	T1	Fabian Coras	2ph	M2	Media
MAP3	ET1	Inicio	PB3	Recopilar requisitos	PO2	Sebastian Huertas	2ph	M3	Baja
MAP4	ET2	Planificación e investigación	PB4	Analisis de requerimiento	T3	Kevin Cari	2ph	M2	Media
MAP5	ET2	Planificación e investigación	PB5	Matriz de Requerimiento	T3	Kevin Cari	2ph	M3	Baja
MAP6	ET2	Planificación e investigación	PB6	Caso de Uso	T3	Kevin Cari	2ph	M1	Alta
MAP7	ET3	Diseño e implementación	PB7	Desarrollo BD Firebase	T1	Fabian Coras	2ph	M3	Baja
MAP8	ET3	Diseño e implementación	PB8	Implementar plantilla de app móvil	T3	Kevin Cari	10ph	M2	Media
MAP9	ET3	Diseño e implementación	PB9	Creación de login y conexión BD Firebase	T2	Sebastian Huertas	4ph	M3	Baja
MAP10	ET3	Diseño e implementación	PB10	Creación de registro y conexión BD Firebase	T2	Sebastian Huertas	4ph	M3	Baja
MAP11	ET3	Diseño e implementación	PB11	CRUD Citas, Categoría en Kotlin	T3	Fabian Coras	5ph	M2	Media
MAP12	ET3	Diseño e implementación	PB12	Elaboración de cronograma de horarios	T3	Sebastian Huertas	4ph	M3	Baja
MAP13	ET3	Diseño e implementación	PB13	Implementación de recomendación de estados salud (IA)	T3	Sebastian Huertas	10ph	M1	Alta
MAP14	ET3	Diseño e implementación	PB14	Creación de citas	T3	Sebastian Huertas	4ph	M3	Baja

## Mapping para la relación de los requerimientos del proyecto (Parte 2)

Cod. MAP	Cod. TI	Etiqueta	Cod. PRO	Nombre activi- dad	Cod. ENC	En car- gado	Esti- mación	Cod. PRI	Prio- ridad
MAP15	ET3	Diseño e implementación	PB15	Cancelación de citas	T3	Sebastian Huertas	4ph	M3	Baja
MAP16	ET3	Diseño e implementación	PB16	Visualización de histórica de citas	T3	Fabian Coras	4ph	M3	Baja
MAP17	ET4	Creación de reportes	PB17	Creación de reportes cotización en PDF	T2	Sebastian Huertas	2ph	M2	Media
MAP18	ET5	Pruebas de CRUD y formularios	PB18	CRUD de Prueba	T3	Fabian Coras	2ph	M3	Baja
MAP19	ET5	Pruebas de CRUD y formularios	PB19	Pruebas de inserción de citas según proveedor BD Firebase	T3	Fabian Coras	2ph	M2	Media
MAP20	ET5	Pruebas de CRUD y formularios	PB20	Prueba de validar registros de usuarios	T3	Fabian Coras	2ph	M2	Media
MAP21	ET5	Pruebas de CRUD y formularios	PB21	Prueba de validar registros de citas	T3	Fabian Coras	2ph	M2	Media

**Actividad 3: Integración del Kanban con el sprint** En esta sesión se define las actividades que se van a desarrollar en el proyecto con el siguiente aplicado con el tablero de Kanban para saber qué actividades aún quedan pendientes.

**Tabla Kanban para el seguimiento**

Código Kanban	Nombre del estado Kanban
KAN1	Solicitudes – Por hacer (To Do)
KAN2	Proceso – En curso (Doing)
KAN3	Verificar – Verificación (Verify)
KAN4	Finalizado – Listo (Done)

**Tabla Sprint con las fechas asignadas para el seguimiento del proyecto**

CODSP	SEM	Inicio	Fin	Observación	Días	Estimación
SP1	2 Sem	10-feb-25	24-feb-25	—	15	6
SP2	2 Sem	24-feb-25	10-mar-25	—	15	6
SP3	5 Sem	10-mar-25	14-abr-25	—	36	51
SP4	3 Sem	14-abr-25	5-may-25	—	22	2
SP5	3 Sem	5-may-25	26-may-25	—	22	8
				Total	110	73

Cuadro 4.9: Sprint para el seguimiento del proyecto (Parte 1)

COD	TBSP	COD SP	COD REQ	COD MAP	NOM BRE ACTIVIDAD	COD ENC	ESTIMACIÓN	COD KAN	Avance %	Observación
1	SP1	RF-001	RF-001	MAP1	Definición del Objetivo del proyecto	SC1	2Ph	KAN4	100 %	
2	SP1	RF-001	RF-001	MAP2	Realizar especificación de funcionalidad	T1	2Ph	KAN4	100 %	
3	SP1	RF-001	RF-001	MAP3	Recopilar requisitos	PO2	2Ph	KAN4	100 %	
4	SP2	RF-002	RF-002	MAP4	Análisis de requerimiento	T3	2Ph	KAN4	100 %	
5	SP2	RF-002	RF-002	MAP5	Matriz de Requerimiento	T3	2Ph	KAN4	100 %	
6	SP2	RF-002	RF-002	MAP6	Caso de Uso	T3	2Ph	KAN4	100 %	
7	SP3	RF-003	RF-003	MAP7	Desarrollo BD Firebase	T1	2Ph	KAN4	100 %	
8	SP3	RF-003	RF-003	MAP8	Implementar plantilla de app móvil	T3	10Ph	KAN4	100 %	
9	SP3	RF-003	RF-003	MAP9	Creación de login y conexión BD Firebase	T2	4Ph	KAN4	100 %	
10	SP3	RF-003	RF-003	MAP10	Creación de registro y conexión BD Firebase	T2	4Ph	KAN4	100 %	
11	SP3	RF-003	RF-003	MAP11	CRUD Citas, Categoría en Kotlin	T1	5Ph	KAN4	100 %	
12	SP3	RF-003	RF-003	MAP12	Elaboración de cronograma de horarios	T2	4Ph	KAN4	100 %	
13	SP3	RF-003	RF-003	MAP13	Implementación recomendación estados salud (IA)	T2	10Ph	KAN4	100 %	
14	SP3	RF-003	RF-003	MAP14	Creación de citas	T2	4Ph	KAN4	100 %	
15	SP3	RF-003	RF-003	MAP15	Cancelación de citas	T2	4Ph	KAN4	100 %	

Cuadro 4.10: Sprint para el seguimiento del proyecto (Parte 2)

COD	TBSP	COD SP	COD REQ	COD MAP	NOM BRE ACTIVIDAD	COD ENC	ESTIMACIÓN	COD KAN	Avance %	Observación
16	SP3	RF-003	RF-003	MAP16	Visualización de historial de citas	T1	4Ph	KAN4	100 %	
17	SP4	RF-004	RF-004	MAP17	Creación de reportes cotización en PDF	T2	2Ph	KAN4	100 %	
18	SP5	RF-005	RF-005	MAP18	CRUD de Prueba	T1	2Ph	KAN4	100 %	
19	SP5	RF-005	RF-005	MAP19	Pruebas de inserción de citas según proveedor BD	T1	2Ph	KAN4	100 %	
20	SP5	RF-005	RF-005	MAP20	Prueba de validar registros de usuarios	T1	2Ph	KAN4	100 %	
21	SP5	RF-005	RF-005	MAP21	Prueba de validar registros de citas	T1	2Ph	KAN4	100 %	

Tabla relacionado con el Sprint 1 para el seguimiento del proyecto

Tablero Kanban - Sprint SP1			
KAN1 Solicitudes	KAN2 Proceso	KAN3 Verificación	KAN4 Finalizado
			Definición del objetivo del proyecto Especificación de funcionalidad Recopilación de requisitos
<b>Avance general:</b>			25 %

Estado de las actividades del Sprint 1 según el tablero Kanban, mostrando el flujo de trabajo y el porcentaje de completado. Las columnas representan las etapas del proceso (KAN1 a KAN4).

Tablero Kanban - Sprint SP2			
KAN1	KAN2	KAN3	KAN4
Solicitudes	Proceso	Verificación	Finalizado
			Análisis de requerimiento Matriz de requerimiento Caso de uso
<b>Avance general:</b>			25 %

*Tablero Kanban correspondiente al Sprint SP2 para el seguimiento del proyecto.*

Tablero Kanban - Sprint SP3			
KAN1	KAN2	KAN3	KAN4
Solicitudes	Proceso	Verificación	Finalizado
			Desarrollo BD Firebase Implementar plantilla de app móvil Creación de login y conexión BD Firebase Creación de registro y conexión BD Firebase CRUD Citas, Categoría en Kotlin Elaboración de cronograma de horarios Creación de citas Cancelación de citas Visualización de historial de citas
<b>Avance general:</b>			25 %

*Tablero Kanban correspondiente al Sprint SP3 para el seguimiento del proyecto.*

#### 4.1.3. Fase 3: Implementacion

En esta fase se llevó a cabo el desarrollo funcional del aplicativo móvil propuesto, integrando las tecnologías seleccionadas durante las etapas de análisis y diseño. Se implementaron todos los módulos descritos en el diseño funcional, utilizando Android Studio, Jetpack Compose para la construcción de interfaces modernas, y Firebase como backend para la gestión de usuarios y base de datos en tiempo real.

Asimismo, se integró una funcionalidad de inteligencia artificial mediante la API de Gemini, la cual permite ofrecer recomendaciones de salud seguras y personalizadas a los usuarios. Esta integración se realizó a través de un ViewModel personalizado encargado de estructurar los prompts médicos y gestionar la comunicación con la IA.

La implementación se organizó en módulos funcionales que abarcan desde el registro de usuarios y la reserva de citas, hasta el panel de administración y el chatbot asistido por IA. Cada módulo fue probado individualmente y en conjunto, asegurando su correcto funcionamiento.

Procedemos a mostrar los sprints para el seguimiento de las actividades.

Sprint 1: 10/03/2025 al 20/04/2025

Tablero Kanban - Sprint SP4			
KAN1	KAN2	KAN3	KAN4
Solicitudes	Proceso	Verificación	Finalizado
			Creación de reportes de cotización en PDF
<b>Avance general:</b>			25 %

Tablero Kanban correspondiente al Sprint SP4 para el seguimiento del proyecto.

Tablero Kanban - Sprint SP5			
KAN1	KAN2	KAN3	KAN4
Solicitudes	Proceso	Verificación	Finalizado
			CRUD de Prueba
			Pruebas de inserción de citas según su proveedor BD Firebase
			Prueba de validar registros de usuarios
			Prueba de validar registros de citas
<b>Avance general:</b>			25 %

Tablero Kanban correspondiente al Sprint SP5 para el seguimiento del proyecto.

### Tablero Sprint 1

The screenshot displays a digital Kanban board with the following structure:

- Header:** Tablero Sprint 1
- Columns:**
  - COMPLETADO:** 4 tasks
  - EN CURSO:** 1 task
  - PENDIENTE:** 1 task
- Task Details:**
  - COMPLETADO:** Definición del alcance de la aplicación móvil (Due Feb 10), Definición de Requerimientos funcionales y no funcionales (Due Feb 15), Investigación y selección de tecnologías (Due Feb 14), Especificación de flujos de procesos (Due Feb 20).
  - EN CURSO:** diseño de estructura de almacenamiento de lista de pacientes y... (Due Feb 22)
  - PENDIENTE:** Desarrollo de la lógica de la IA (Due Feb 28)
- Buttons:** + Agregar Tarea (Add Task) button at the bottom of each column.

Nota: Elaboración Propia.

Sprint 2: 20/03/2025 al 14/04/2025

### Tablero Sprint 2

The screenshot shows a digital Kanban board titled "Tablero". The board is organized into three main columns: "COMPLETADO" (Completed), "EN CURSO" (In Progress), and "PENDIENTE" (Pending). Each column has a count of items (5, 0, and 1 respectively) and a "Agregar Tarea" (Add Task) button.

- COMPLETADO (Completed):** Contains one task: "Integración del sistema de consultas con IA en la app". It includes a due date of "Mar 20" and standard edit/delete icons.
- EN CURSO (In Progress):** Contains one task: "Reservas de citas". It includes a due date of "Mar 23" and standard edit/delete icons.
- PENDIENTE (Pending):** Contains one task: "Pruebas de carga y rendimiento". It includes a due date of "Apr 15" and standard edit/delete icons.

Below the columns, there are four additional boxes representing tasks:

- "Implementación de seguridad de datos" (Due Mar 29)
- "Panel de administración" (Due Apr 5)
- "Gestión de Terapeutas" (Due Apr 14)
- A general "Agregar Tarea" (Add Task) button.

Nota: Elaboración Propia.

Sprint 3: 20/03/2025 al 04/04/2025

### Tablero Sprint 3

The scrum board displays the following information:

- COMPLETADO (Green):** 3 tasks. Subareas: **Lista de Terapeutas** (due Apr 21), **Lista de Pacientes** (due Apr 23), and **Horarios** (due Apr 24). Each has a 'Agregar Tarea' button.
- EN CURSO (Purple):** 0 tasks. Buttons: '+ Agregar Tarea'.
- PENDIENTE (Grey):** 0 tasks. Buttons: '+ Agregar Tarea'.

Nota: Elaboración Propia.

Sprint 4: 21/04/2025 al 23/04/2025

### Tablero Sprint 4

The scrum board displays the following information:

- COMPLETADO (Green):** 2 tasks. Subarea: **Actualización y mantenimiento constante** (due Apr 21). Includes a checklist icon, a checkmark, a plus sign, a circular arrow, and a delete icon. Each has a 'Agregar Tarea' button.
- EN CURSO (Purple):** 0 tasks. Buttons: '+ Agregar Tarea'.
- PENDIENTE (Grey):** 0 tasks. Buttons: '+ Agregar Tarea'.

Nota: Elaboración Propia.

## 4.2. Módulos Implementados

A continuación, se presentan los módulos desarrollados en la aplicación móvil, detallando sus funcionalidades, tecnologías empleadas y su relación directa con la inteligencia artificial implementada.

### 4.2.1. Módulo 1: Generar la API de gemini en Google AI Studio

- Obtención de la API Key desde Google Cloud
- Entendiendo la Petición a la API de Gemini

#### Obtención de la API Key desde Google Cloud

#### Pasos para obtener la API Key de Gemini 1.5 Flash

Para poder utilizar el modelo **Gemini 1.5 Flash** de Google, es necesario obtener una clave de API siguiendo los pasos a continuación:

##### 1. Acceder a Google AI Studio:

Visitar el sitio web <https://makersuite.google.com/app> utilizando una cuenta de Google.

##### 2. Activar la API de Gemini (Generative Language API):

- Dentro del proyecto creado, buscar y habilitar la API llamada **Generative Language API**.

##### 3. Crear una clave API:

- Ir a **Claves API**.
- Hacer clic en **Crear Clave de API**.
- Guardar la clave generada, ya que será utilizada dentro del código de Android Studio para realizar peticiones al modelo Gemini.

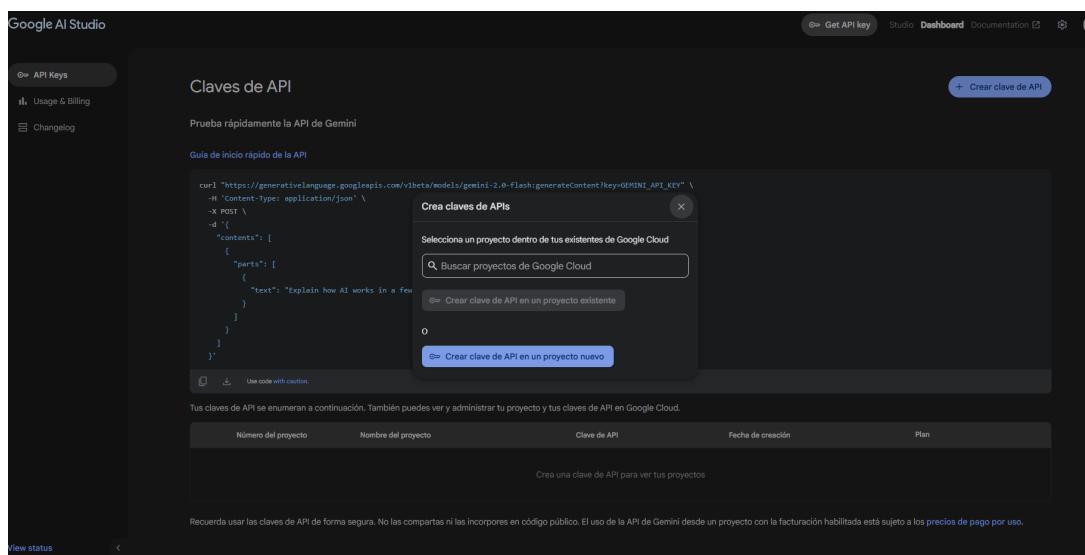


Figura 4.1: Crear clave de API

## Entendiendo la Petición a la API de Gemini

Antes de integrarlo en Android Studio, es clave entender cómo funciona una petición a Gemini. La estructura básica es:

```
curl "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent?key=GEMINI_API_KEY" \
-H 'Content-Type: application/json' \
-X POST \
-d '{
  "contents": [
    {
      "parts": [
        {
          "text": "Explain how AI works in a few words"
        }
      ]
    }
}'
```

Figura 4.2: Crear clave de API

### Explicación Técnica: Comunicación con Gemini 1.5 Flash mediante API REST

Para establecer la comunicación entre la aplicación móvil y el modelo de inteligencia artificial Gemini, se utiliza un enfoque basado en solicitudes HTTP mediante la herramienta `curl` o desde bibliotecas HTTP en Android.

- **curl:** Es una herramienta de línea de comandos que permite realizar peticiones HTTP a servicios externos. En este contexto, se usa para enviar el *prompt* del usuario a la API de Gemini y recibir la respuesta generada.
- **URL base:**

El endpoint que permite interactuar con el modelo Gemini 1.5 Flash es:

```
generativelanguage.googleapis.com
/v1beta/models/gemini-2.0-flash:generateContent?key=GEMINI_APKEY
```

Este endpoint representa la dirección a la cual se envían los datos para que el modelo procese la entrada.

- **Clave de API:**  
El parámetro `key=GEMINI_API_KEY` corresponde a la clave de autenticación que se obtiene desde Google Cloud Console. Esta clave valida que la aplicación esté autorizada para consumir el servicio de IA.
- **"text":**  
Este campo representa el *prompt* o entrada que el usuario envía al modelo. Es el contenido que el modelo utilizará como base para generar una respuesta.
- **contents":**  
Se refiere al cuerpo completo de la conversación, incluyendo instrucciones, contexto o historial. Este campo es fundamental para mantener coherencia en las interacciones y generar respuestas precisas.

- **Respuesta del modelo:**

Una vez procesada la entrada, la API devuelve una salida en formato JSON que contiene el texto generado por Gemini. Este texto es interpretado por la app móvil y mostrado al usuario como una recomendación clínica personalizada.

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.				
Project number	Project name	API key	Created	Plan
...3190	Gemini API	...OpBE	May 15, 2025	Free Set up Billing View usage data

Figura 4.3: Pantalla de Inicio

#### 4.2.2. Módulo 2: Pantallas de Inicio, Login y Registro

- **Tecnología:** Firebase Authentication

- **Descripción:**

- Registro y autenticación mediante correo y contraseña.
- Validación de formularios y recuperación de contraseña.

#### Clases

```
package es.shch.gemini2

import ...

class BienvenidaActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_bienvenida)

        val tvWelcome = findViewById<TextView>(R.id.tvWelcome)
        val btnLogin = findViewById<Button>(R.id.btnLogin)
        val btnRegister = findViewById<Button>(R.id.btnRegister)

        // Configurar el mensaje de bienvenida
        tvWelcome.text = "¡Bienvenida a ReneApp!"

        // Botón para ir a la pantalla de inicio de sesión
        btnLogin.setOnClickListener {
            startActivity(Intent(packageContext, LoginActivity::class.java))
        }

        // Botón para ir a la pantalla de registro
        btnRegister.setOnClickListener {
            startActivity(Intent(packageContext, RegisterActivity::class.java))
        }
    }
}
```

Figura 4.4: Interfaz de bienvenidaActivity

- **bienvenidaActivity.kt:** Pantalla inicial de la aplicación que muestra un mensaje de bienvenida y proporciona acceso a las pantallas de inicio de sesión y registro de nuevos usuarios mediante botones dedicados.

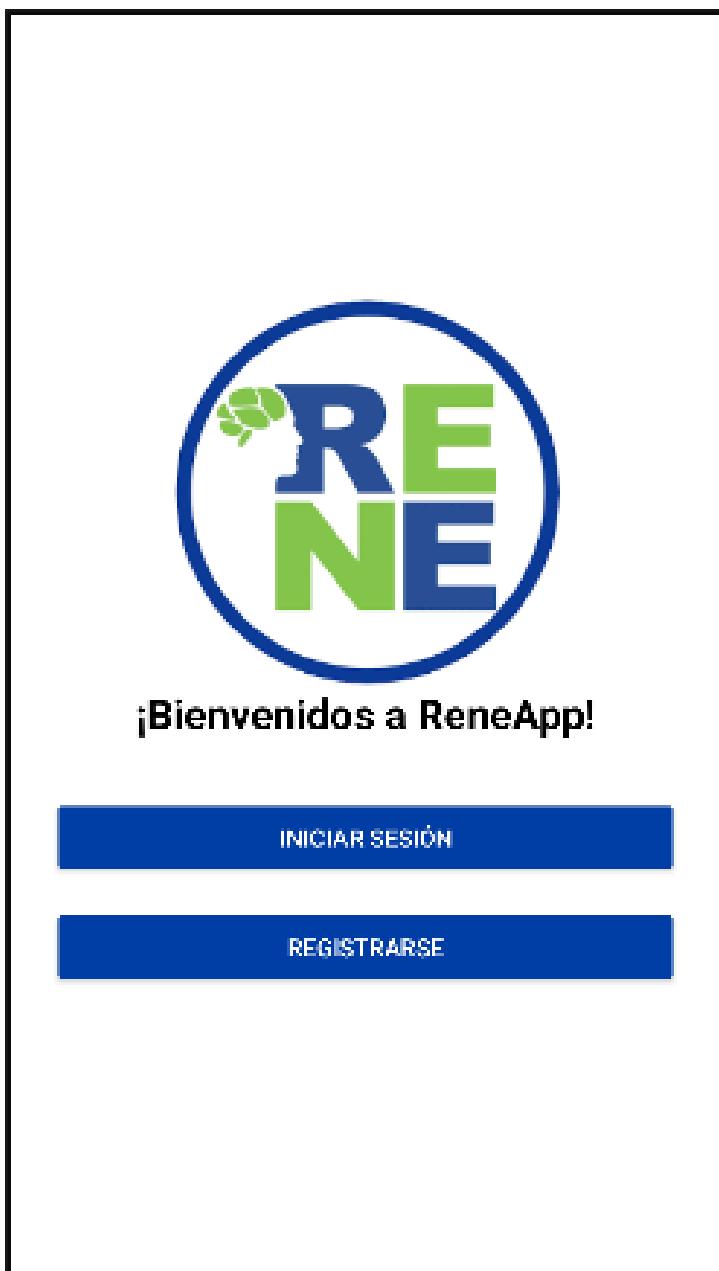


Figura 4.5: Pantalla de Inicio

```

class LoginActivity : AppCompatActivity() {

    // Credenciales fijas para el admin
    private val ADMIN_EMAIL = "RenelNeuro@clinics.com"
    private val ADMIN_PASSWORD = "2025Renapp"

    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        auth = FirebaseAuth.getInstance()

        val btnLogin = findViewById(R.id.btnLoginAction)
        val etEmail = findViewById<EditText>(R.id.etEmailLogin)
        val etPassword = findViewById<EditText>(R.id.etPasswordLogin)

        btnLogin.setOnClickListener {
            val email = etEmail.text.toString().trim()
            val password = etPassword.text.toString().trim()

            when {
                email.isEmpty() || password.isEmpty() -> {
                    showMessage("Complete todos los campos")
                }
                email == ADMIN_EMAIL && password == ADMIN_PASSWORD -> {
                    startActivity(Intent(packageContext, AdminActivity::class.java))
                    finish()
                }
                else -> {
                    loginPaciente(email, password)
                }
            }
        }
    }
}

```

Figura 4.6: Clase LoginActivity

- **LoginActivity.kt:** Pantalla de autenticación con validación de campos. Incluye credenciales predefinidas para acceso administrativo y redirección diferenciada para administradores (AdminActivity) y pacientes regulares.

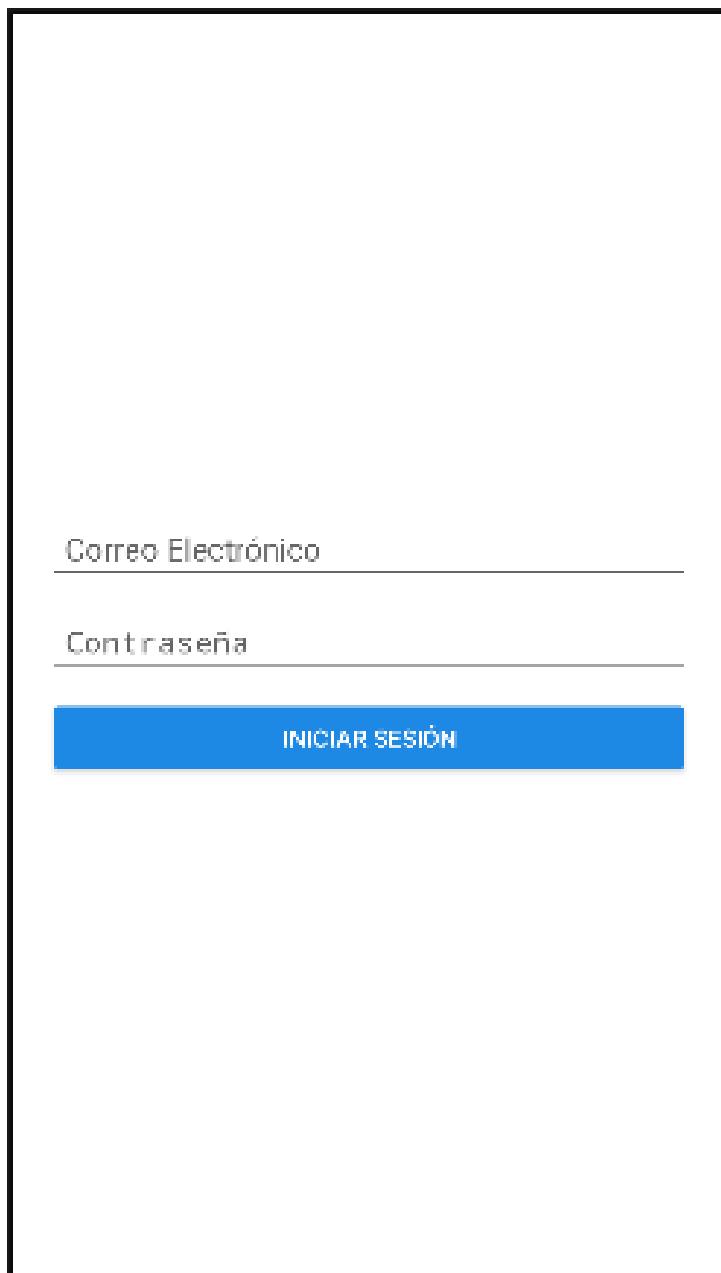
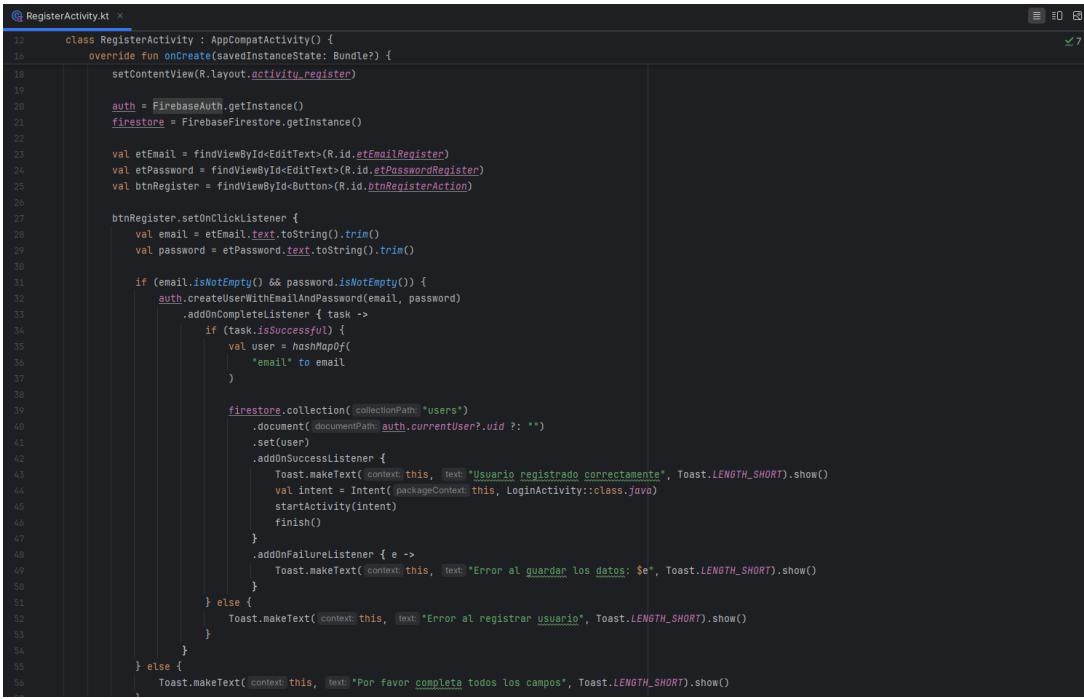


Figura 4.7: Pantalla de Login



```
class RegisterActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        setContentView(R.layout.activity_register)
        auth = FirebaseAuth.getInstance()
        firestore = FirebaseFirestore.getInstance()

        etEmail = findViewById<EditText>(R.id.etEmailRegister)
        etPassword = findViewById<EditText>(R.id.etPasswordRegister)
        btnRegister = findViewById<Button>(R.id.btnRegisterAction)

        btnRegister.setOnClickListener {
            val email = etEmail.text.toString().trim()
            val password = etPassword.text.toString().trim()

            if (email.isNotEmpty() && password.isNotEmpty()) {
                auth.createUserWithEmailAndPassword(email, password)
                    .addOnCompleteListener { task ->
                        if (task.isSuccessful) {
                            val user = hashMapOf(
                                "email" to email
                            )

                            firestore.collection("users")
                                .document(documentPath(auth.currentUser?.uid ?: ""))
                                .set(user)
                                .addOnSuccessListener {
                                    Toast.makeText(context, "Usuario registrado correctamente", Toast.LENGTH_SHORT).show()
                                    val intent = Intent(packageContext, LoginActivity::class.java)
                                    startActivity(intent)
                                    finish()
                                }
                                .addOnFailureListener { e ->
                                    Toast.makeText(context, "Error al guardar los datos: $e", Toast.LENGTH_SHORT).show()
                                }
                        } else {
                            Toast.makeText(context, "Error al registrar usuario", Toast.LENGTH_SHORT).show()
                        }
                    }
            } else {
                Toast.makeText(context, "Por favor completa todos los campos", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Figura 4.8: Clase RegisterActivity

- **RegisterActivity.kt:** Registro de nuevos usuarios con validación de campos. Almacena email en Firestore después de crear la autenticación en Firebase Auth, con redirección automática al login tras registro exitoso.

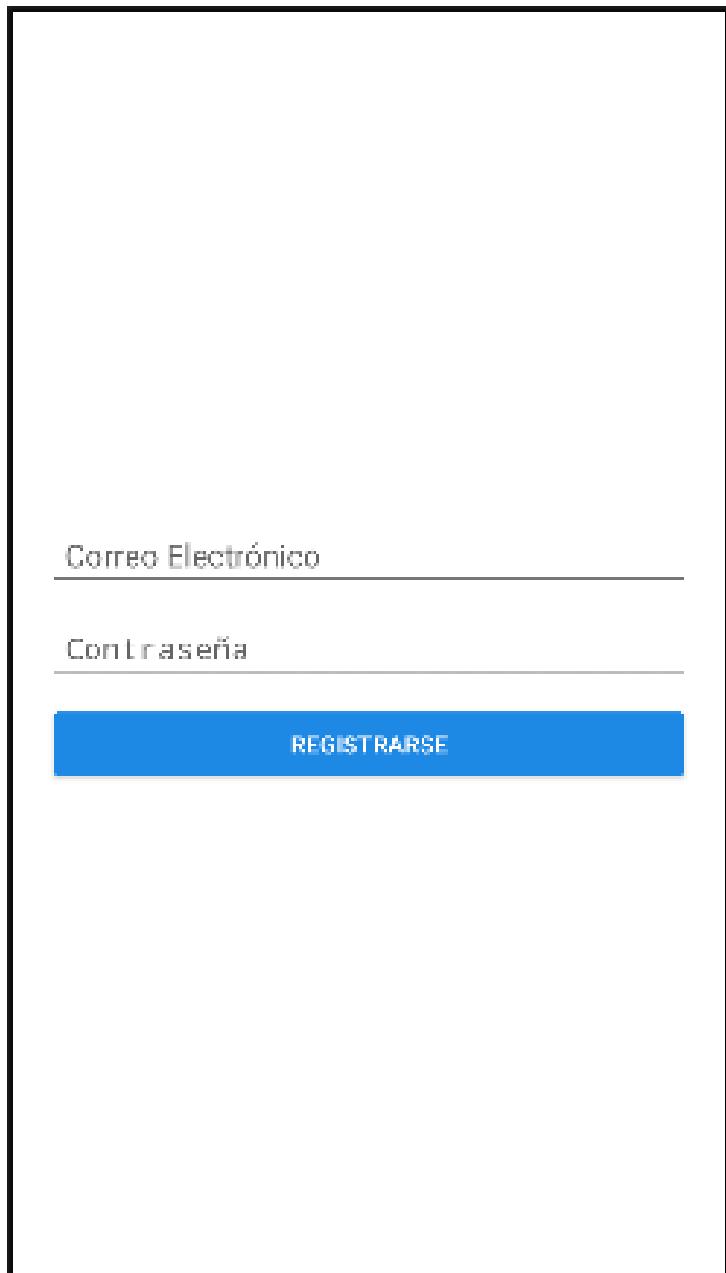


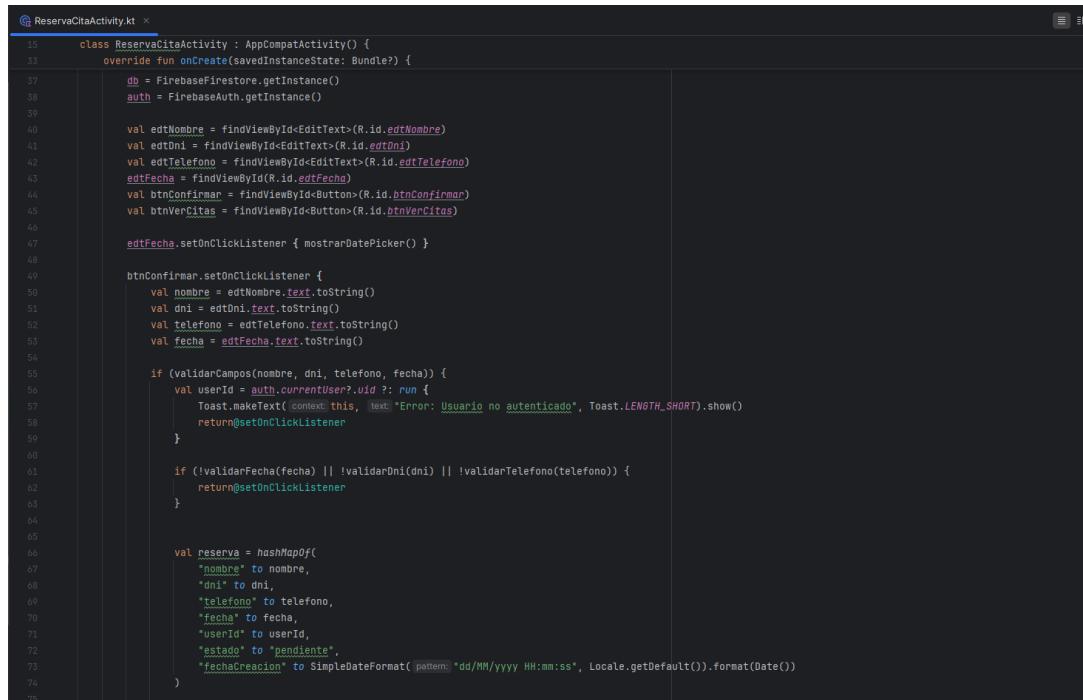
Figura 4.9: Pantalla de Registro

#### 4.2.3. Módulo 3: Reserva y Visualización de Citas

- **Tecnología:** Firebase Realtime Database

- **Descripción:**

- Integración de DatePicker y TimePicker para la selección de fecha y hora.
- Almacenamiento de las citas con nombre, correo y detalles en tiempo real.
- Listado de citas en la sección “Mis Citas”.



```
ReservaCitaActivity.kt
15  class ReservaCitaActivity : AppCompatActivity() {
16      override fun onCreate(savedInstanceState: Bundle?) {
17          db = FirebaseFirestore.getInstance()
18          auth = FirebaseAuth.getInstance()
19
20          val edtNombre = findViewById<EditText>(R.id.edtNombre)
21          val edtDni = findViewById<EditText>(R.id.edtDni)
22          val edtTelefono = findViewById<EditText>(R.id.edtTelefono)
23          edtFecha = findViewById(R.id.edtFecha)
24          val btnConfirmar = findViewById<Button>(R.id.btnConfirmar)
25          val btnVerCitas = findViewById<Button>(R.id.btnVerCitas)
26
27          edtFecha.setOnClickListener { mostrarDatePicker() }
28
29          btnConfirmar.setOnClickListener {
30              val nombre = edtNombre.text.toString()
31              val dni = edtDni.text.toString()
32              val telefono = edtTelefono.text.toString()
33              val fecha = edtFecha.text.toString()
34
35              if (validarCampos(nombre, dni, telefono, fecha)) {
36                  val userId = auth.currentUser?.uid ?: run {
37                      Toast.makeText(context, "Error: Usuario no autenticado", Toast.LENGTH_SHORT).show()
38                      return@setOnClickListener
39                  }
40
41                  if (!validarFecha(fecha) || !validarDni(dni) || !validarTelefono(telefono)) {
42                      return@setOnClickListener
43                  }
44
45                  val reserva = hashMapOf(
46                      "nombre" to nombre,
47                      "dni" to dni,
48                      "telefono" to telefono,
49                      "fecha" to fecha,
50                      "userId" to userId,
51                      "estado" to "pendiente",
52                      "fechaCreacion" to SimpleDateFormat(pattern: "dd/MM/yyyy HH:mm:ss", Locale.getDefault()).format(Date())
53                  )
54
55                  FirebaseFirestore.getInstance().collection("reservas").add(reserva)
56
57                  Toast.makeText(context, "Reserva creada exitosamente", Toast.LENGTH_SHORT).show()
58
59                  finish()
60              }
61          }
62
63      }
64
65      private fun validarCampos(nombre: String, dni: String, telefono: String, fecha: String): Boolean {
66          return nombre.isNotEmpty() && dni.isNotEmpty() && telefono.isNotEmpty() && fecha.isNotEmpty()
67      }
68
69      private fun validarDni(dni: String): Boolean {
70          return dni.length == 9 && dni.toIntOrNull() != null
71      }
72
73      private fun validarTelefono(telefono: String): Boolean {
74          return telefono.length == 9 && telefono.toIntOrNull() != null
75      }
76
77      private fun validarFecha(fecha: String): Boolean {
78          return fecha.length == 10 && fecha.toIntOrNull() != null
79      }
80
81      private fun mostrarDatePicker() {
82          val calendar = Calendar.getInstance()
83          val datePicker = DatePickerDialog(this, R.style.DialogTheme, { _, year, month, dayOfMonth ->
84              calendar.set(Calendar.YEAR, year)
85              calendar.set(Calendar.MONTH, month)
86              calendar.set(Calendar.DAY_OF_MONTH, dayOfMonth)
87              val date = calendar.time
88              edtFecha.setText(date.toString())
89          }, calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH), calendar.get(Calendar.DAY_OF_MONTH))
90          datePicker.show()
91      }
92
93  }
```

Figura 4.10: Clase ReservaCitaActivity

- **ReservaCitaActivity.kt:** Formulario completo para reserva de citas médicas con validación de DNI, teléfono y fecha. Almacena los datos en Firestore incluyendo marca temporal y estado "pendiente" por defecto.

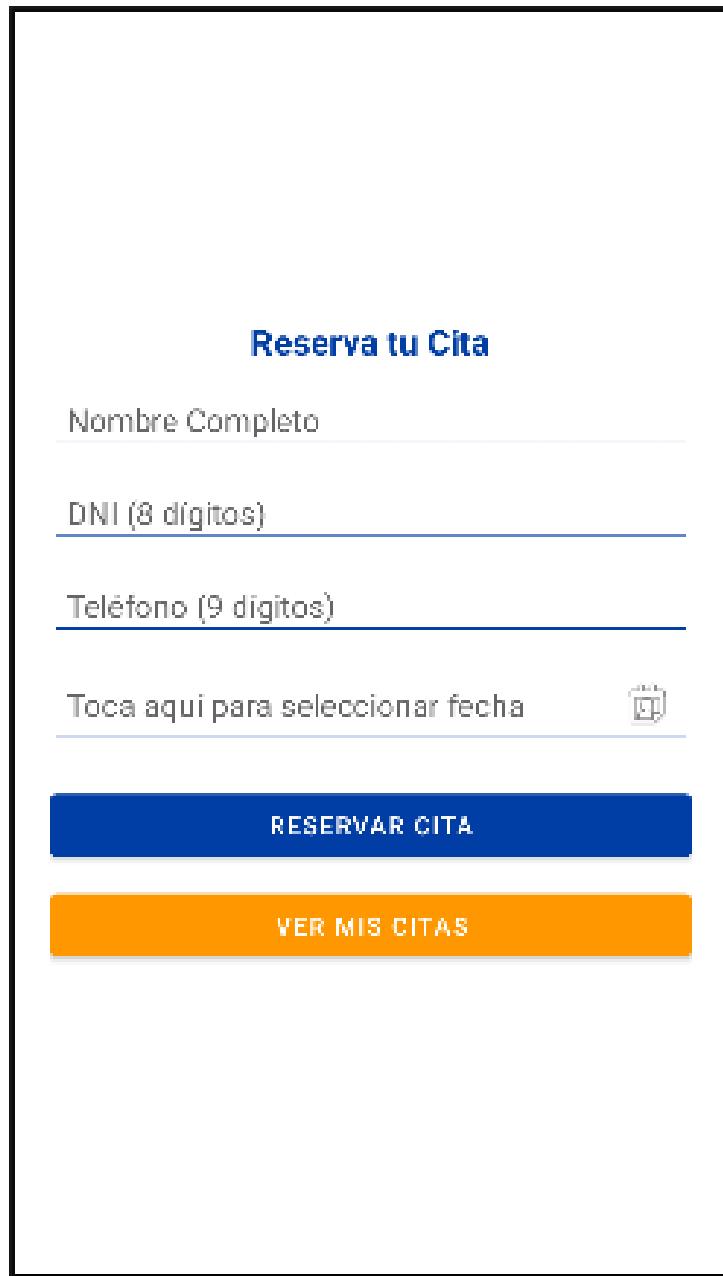


Figura 4.11: Pantalla para reservar una cita

Figura 4.12: Clase CitaAdapter

- **CitaAdapter.kt**: Adaptador personalizado para RecyclerView que muestra la lista de citas médicas. Gestiona la visualización de datos del paciente (nombre, DNI, teléfono), fecha/hora de la cita, terapeuta asignado y estado. Incluye botones para acciones como eliminar citas.

```
17 class MisCitasActivity : AppCompatActivity() {
18 }
19
20 private fun cargarCitas() {
21     val userId = auth.currentUser?.uid ?: return
22
23     db.collection(collectionPath: "reservas")
24         .whereEqualTo("field:userid", userId)
25         .get()
26         .addOnSuccessListener { documents ->
27             citasList.clear()
28             citasList.addAll(documents.documents)
29             citaAdapter.actualizarCitas(citasList)
30
31             if (citasList.isEmpty()) {
32                 txtSinCitas.visibility = View.VISIBLE
33                 recyclerCitas.visibility = View.GONE
34             } else {
35                 txtSinCitas.visibility = View.GONE
36                 recyclerCitas.visibility = View.VISIBLE
37             }
38         }
39         .addOnFailureListener { e ->
40             Toast.makeText(context: this, text: "Error al cargar citas: ${e.message}", Toast.LENGTH_SHORT).show()
41             txtSinCitas.visibility = View.VISIBLE
42             recyclerCitas.visibility = View.GONE
43         }
44     }
45
46 private fun mostrarDialogoConfirmacion(cita: DocumentSnapshot) {
47     AlertDialog.Builder(this)
48         .setTitle("Eliminar Cita")
49         .setMessage("¿Estás seguro que deseas eliminar esta cita?")
50         .setPositiveButton(text: "Sí") { dialog, _ ->
51             eliminarCita(cita)
52             dialog.dismiss()
53         }
54         .setNegativeButton(text: "No") { dialog, _ ->
55             dialog.dismiss()
56         }
57         .create()
58         .show()
59 }
```

Figura 4.13: Item de Lista de Citas

- **Item de Cita:** Elemento de lista interactivo que muestra información detallada

de cada cita (nombre, DNI, teléfono y fecha). Diseñado con efecto de selección y disposición vertical clara.

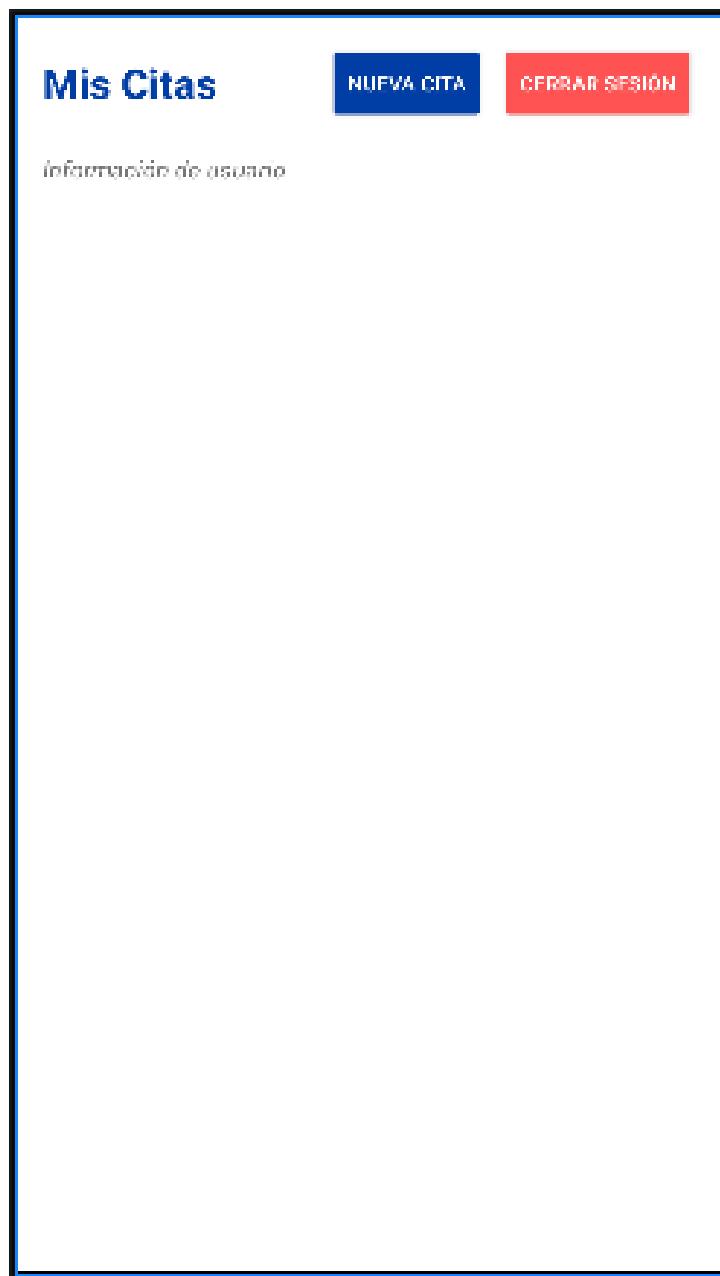


Figura 4.14: Visualización de citas registradas

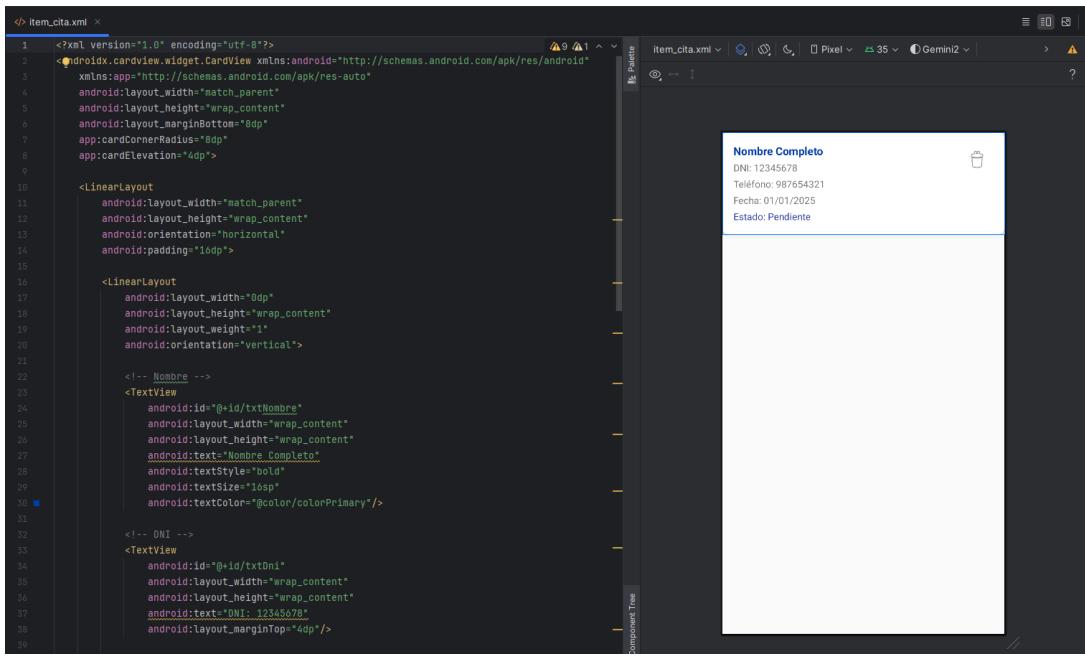


Figura 4.15: CardView de Paciente

- **Card de Paciente:** Tarjeta con información médica que muestra nombre completo y DNI del paciente. Diseño con bordes redondeados, sombra y disposición horizontal para mostrar datos clave.

#### 4.2.4. Módulo 4: Implementación de la IA con Gemini en Android Studio

### Requisitos Previos y Configuración del Proyecto Android

Para implementar correctamente la API de Gemini en una aplicación Android, es necesario cumplir con los siguientes requisitos previos y realizar una configuración inicial adecuada en el entorno de desarrollo.

#### Requisitos Previos

Este instructivo asume conocimientos básicos en el uso de Android Studio para el desarrollo de aplicaciones móviles. Antes de comenzar, asegúrese de que su entorno y su proyecto cumplan con lo siguiente:

- **Android Studio:** Tener instalada la versión más reciente del entorno de desarrollo.
- **Nivel de API:** Su aplicación Android debe estar configurada para orientarse al nivel de API 21 (Android 5.0 Lollipop) o superior.

#### Configura tu Proyecto Android

Antes de realizar llamadas a la API de Gemini, es necesario preparar su proyecto con los siguientes pasos:

##### 1. Configurar la clave de API:

Obtenga la clave de API desde Google Cloud Console, como se explicó previamente, y guárdela en un archivo seguro (En, `local.properties`) o como una constante protegida dentro del código.

```
val apiKey = BuildConfig.apiKey
```

##### 2. Agregar dependencias del SDK:

En el archivo `build.gradle` (Module: app), incluya las dependencias necesarias para realizar solicitudes HTTP (como `Retrofit`, `OkHttp` o `HttpURLConnection`), además de bibliotecas para parseo JSON si lo requiere (como `Gson`).

```
dependencies {
    // ... other androidx dependencies

    // add the dependency for the Google AI client SDK for Android
    implementation("com.google.ai.client.generativeai:generativeai:0.9.0")
}
```

##### 3. Inicializar el modelo Gemini:

Antes de realizar llamadas a las APIs, debes inicializar el modelo generativo:

```
val generativeModel = GenerativeModel(
    // The Gemini 1.5 models are versatile and work with most use cases
    modelName = "gemini-1.5-flash",
    // Access your API key as a Build Configuration variable (see "Set up your API key" above)
    apiKey = BuildConfig.apiKey
)
```

Una vez completada esta configuración, su aplicación estará lista para interactuar con el modelo **Gemini 1.5 Flash**, permitiendo la generación de recomendaciones clínicas personalizadas en tiempo real.

## Agregar la Dependencia del SDK de Gemini

Para habilitar el uso del modelo de inteligencia artificial Gemini dentro de tu aplicación Android, es necesario agregar la dependencia correspondiente al SDK de Google para Android.

- Abre el archivo de configuración de Gradle de tu módulo de aplicación, ubicado en:  
`<project>/<app-module>/build.gradle.kts`

A continuación, agrega la siguiente línea dentro del bloque `dependencies{}`:

```
dependencies {
    implementation("com.google.ai.client:generativelanguage:0.4.0")
}
```

Esta dependencia permite realizar llamadas al modelo Gemini 1.5 Flash desde tu aplicación Android, facilitando la generación de contenido utilizando la API de lenguaje generativo de Google.

**Nota:** IMPORTANTE sincronizar el proyecto después de modificar el archivo `build.gradle.kts`, para que la dependencia sea descargada e integrada correctamente.

## Inicialización del Modelo Generativo

Antes de realizar llamadas a la API de Gemini, es necesario inicializar el modelo generativo desde tu aplicación Android. A continuación, se muestra el ejemplo en **Kotlin** utilizando la clase `GenerativeModel` proporcionada por el SDK de Google:

Listing 4.1: Inicialización del modelo Gemini 1.5 Flash en Kotlin

```
val generativeModel = GenerativeModel(
    // El modelo Gemini 1.5 Flash es versatil y adecuado para la
    // mayoría de los casos de uso
    modelName = "gemini-1.5-flash",
    // Se accede a la API key desde una variable de configuración
    // del proyecto
    apiKey = BuildConfig.apiKey
)
```

Este fragmento crea una instancia del modelo `gemini-1.5-flash`, que será utilizado para generar recomendaciones clínicas personalizadas a partir de la entrada textual del paciente.

### Consideración Técnica:

- Es importante seleccionar un modelo apropiado según el tipo de entrada y el caso de uso.
- En este proyecto se utiliza `gemini-1.5-flash`, un modelo optimizado para **entradas multimodales**, con baja latencia y alta capacidad de respuesta, ideal para aplicaciones móviles.

## 4.3. Implementación del Chat de Consultas con Gemini

### 1.5 Flash

#### Visión General

El chat de consultas fue implementado mediante el modelo **Gemini 1.5 Flash**, desarrollado por Google DeepMind. Este modelo permite generar recomendaciones terapéuticas automáticas en tiempo real a partir de texto libre ingresado por el paciente. Gracias a su capacidad de mantener el contexto conversacional y admitir entradas multimodales, el sistema ofrece respuestas clínicas relevantes en función del historial de mensajes.

#### Casos de Uso Implementados

A través de la API de Gemini y su SDK oficial para Android, se implementaron los siguientes escenarios:

1. Generación de texto desde entradas de solo texto:

```
val generativeModel = GenerativeModel(  
    modelName = "gemini-1.5-flash",  
    apiKey = BuildConfig.apiKey  
)  
  
val prompt = "Tengo dolor en la espalda baja y estoy en  
rehabilitación"  
val response = generativeModel.generateContent(prompt)  
println(response.text)
```

2. Generación de texto desde texto (Multimodal):

```
val inputContent = content {  
    image(bitmap1)  
    text(" Que ejercicios me recomiendas para esta lesión?")  
}  
val response = generativeModel.generateContent(inputContent)  
println(response.text)
```

3. Conversación de múltiples turnos (Chat):

```
val chat = generativeModel.startChat(  
    history = listOf(  
        content(role = "user") { text("Hola, me duele la  
            pierna izquierda") },  
        content(role = "model") { text(" Desde cundo tienes  
            esa molestia?") }  
    )  
)  
chat.sendMessage("Desde hace tres días, sobre todo al caminar  
")
```

4. Streaming para respuestas más rápidas:

```

var fullResponse = ""
generativeModel.generateContentStream(prompt).collect { chunk
    ->
    print(chunk.text)
    fullResponse += chunk.text
}

```

## 5. Control de parámetros del modelo:

```

val config = generationConfig {
    temperature = 0.9f
    topK = 16
    topP = 0.1f
    maxOutputTokens = 200
}
val generativeModel = GenerativeModel(
    modelName = "gemini-1.5-flash",
    apiKey = BuildConfig.apiKey,
    generationConfig = config
)

```

## 6. Configuración de seguridad (bloqueo de contenido dañino):

```

val generativeModel = GenerativeModel(
    modelName = "gemini-1.5-flash",
    apiKey = BuildConfig.apiKey,
    safetySettings = listOf(
        SafetySetting(HarmCategory.HARASSMENT, BlockThreshold.
            ONLY_HIGH),
        SafetySetting(HarmCategory.HATE_SPEECH, BlockThreshold
            .MEDIUM_AND_ABOVE)
    )
)

```

## 7. Conteo de tokens para controlar la longitud de entrada:

```

val (totalTokens) = generativeModel.countTokens("Describe
ejercicios para fortalecer rodilla")

```

## Importancia en el Flujo Clínico

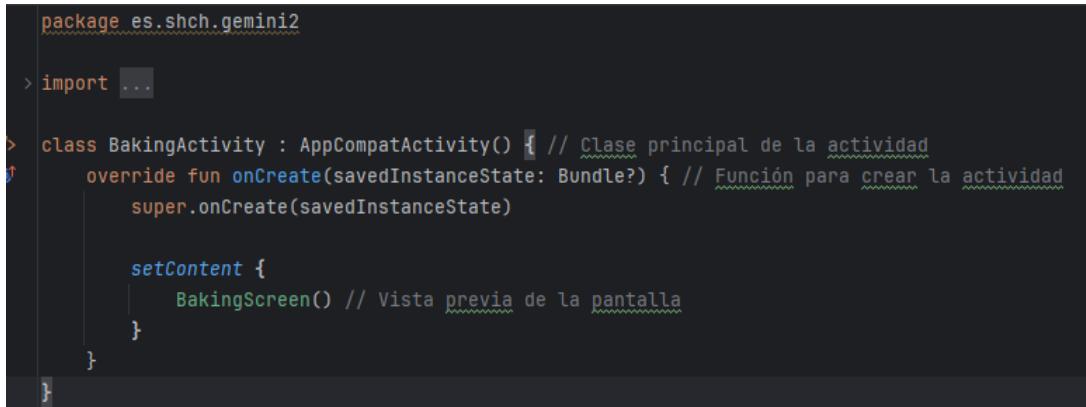
El sistema de chat permite al paciente describir sus síntomas en lenguaje natural. Esta entrada es enviada como prompt al modelo Gemini, el cual devuelve una recomendación terapéutica. La respuesta puede incluir:

- Ejercicios sugeridos.
- Consejos para aliviar molestias.
- Alertas para consultar presencialmente si los síntomas persisten.

Esta arquitectura reduce la carga operativa de los terapeutas y garantiza que el paciente reciba atención incluso fuera del horario clínico, cumpliendo el objetivo de acompañamiento continuo del tratamiento.

#### 4.3.1. Creacion de las clases e Interfaces para la consulta IA

**BakingActivity:** Es la clase principal que lanza la interfaz de la IA. Se encarga de iniciar la actividad y renderizar la vista ‘BakingScreen’, que contiene la interacción con el modelo Gemini. Funciona como el punto de entrada a la pantalla de IA dentro de la app.



```
package es.shch.gemini2

> import ...

> class BakingActivity : AppCompatActivity() { // Clase principal de la actividad
    > override fun onCreate(savedInstanceState: Bundle?) { // Función para crear la actividad
        super.onCreate(savedInstanceState)

        setContentView {
            BakingScreen() // Vista previa de la pantalla
        }
    }
}
```

Figura 4.16: Código del BakingActivity

#### Componente de Interfaz: BakingScreen

BakingScreen representa la interfaz de usuario principal para la interacción con el modelo de inteligencia artificial Gemini dentro de la aplicación Android. Esta pantalla permite al usuario seleccionar una imagen, escribir una solicitud textual (*prompt*) y enviar ambos elementos al modelo para recibir una respuesta generada.

#### Funcionalidad Principal

El componente está diseñado para manejar entradas multimodales (imagen + texto) y mostrar resultados dinámicos en pantalla. La respuesta generada se presenta en formato de texto enriquecido utilizando **Markdown**, lo cual mejora la legibilidad y presentación del contenido.

#### Elementos que incluye

- **Galería de imágenes clicables:**

Una fila horizontal con miniaturas de imágenes seleccionables. Las imágenes elegidas pueden ser utilizadas como entrada visual junto con el texto al enviar la solicitud al modelo Gemini.

- **Campo de texto para el *prompt*:**

Permite al usuario escribir una instrucción o pregunta en lenguaje natural que será procesada por la IA junto con las imágenes (si se incluyen).

- Botón de envío:

Inicia el proceso de generación llamando al método `generateContent()` con los datos proporcionados por el usuario.

#### ■ Visualización de resultados:

Muestra en tiempo real la respuesta generada por la IA o un mensaje de error, según el estado de ejecución gestionado mediante el patrón **UiState** (**Loading**, **Success**, **Error**).

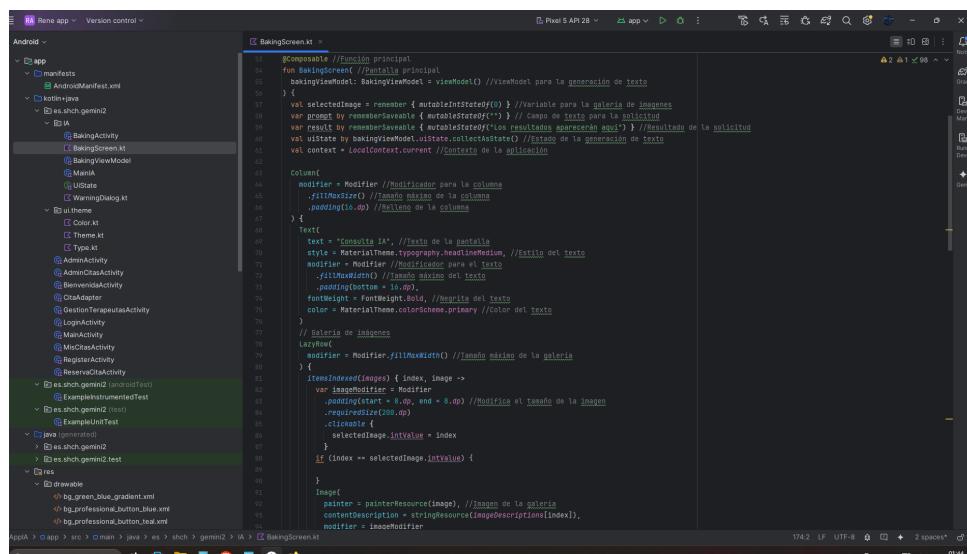


Figura 4.17: Código del BackingScreen

## Importancia

Este componente proporciona una experiencia de usuario intuitiva, adaptada a dispositivos móviles, y actúa como punto de integración entre el paciente y el sistema inteligente. Facilita una comunicación natural con la IA para generar recomendaciones terapéuticas de forma contextualizada y visual.

**Componente de Lógica:** BakingViewModel

`BakingViewModel` es el componente de arquitectura encargado de manejar la lógica de negocio asociada a la interacción con el modelo de inteligencia artificial **Gemini 1.5 Flash**. Este *ViewModel*, parte del patrón MVVM (Model-View-ViewModel), actúa como intermediario entre la interfaz de usuario (`BakingScreen`) y la capa de procesamiento de datos.

## **Responsabilidades principales**

- Gestiona la preparación y envío de solicitudes al modelo Gemini.
  - Controla el estado de la respuesta mediante un sistema de estados definido como `UiState`.
  - Expone datos observables que permiten a la interfaz de usuario reaccionar dinámicamente a los resultados de la IA.

## Función: sendPrompt(bitmap, prompt)

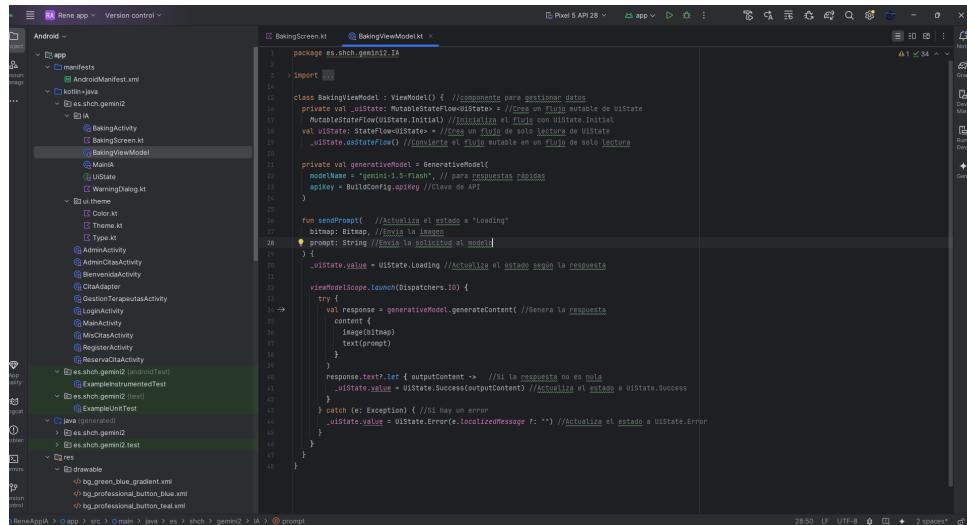
Esta función es responsable de ejecutar la llamada asincrónica al modelo generativo con entrada multimodal. Se compone de los siguientes pasos:

1. Establece el estado en `UiState.Loading` mientras se realiza el procesamiento.
2. Envía la imagen (`bitmap`) y el texto (`prompt`) al modelo `gemini-1.5-flash` usando el método `generateContent()`.
3. Si la respuesta se obtiene exitosamente, actualiza el estado a `UiState.Success`, conteniendo el texto generado por la IA.
4. En caso de error (fallo de red, error del modelo o datos inválidos), establece el estado en `UiState.Error`.

## Gestión de Estados

La clase utiliza un sistema de estados definidos como:

- `UiState.Loading`: Se muestra un indicador de carga en la interfaz mientras se procesa la solicitud.
- `UiState.Success`: Muestra el resultado generado por el modelo Gemini.
- `UiState.Error`: Muestra un mensaje de error explicativo ante fallos en la generación o conectividad.



The screenshot shows the Android Studio interface with the code editor open to the `BakingViewModel.kt` file. The code implements a `BakingViewModel` class that interacts with a `GenerativeModel` to process prompts and return generated responses. It uses `UiState` enum values to manage the application state (Loading, Success, Error).

```
package es.sch.gemini2.IA

import ...

class BakingViewModel : ViewModel() {
    private val _uiState: MutableLiveData<UiState> = MutableLiveData<UiState>(UiState.Initial)
    private val _generatedText: MutableLiveData<String> = MutableLiveData<String>("")

    fun sendPrompt(bitmap: Bitmap, prompt: String) {
        _uiState.value = UiState.Loading //Actualiza el estado a "Loading"
        generativeModel.generateContent(bitmap, prompt) //Envía la imagen y el texto al modelo
    }

    fun getGeneratedText(): LiveData<String> {
        return _generatedText
    }
}

enum class UiState {
    Initial,
    Loading,
    Success(String),
    Error(String)
}
```

Figura 4.18: Código del BakingViewModel

## Importancia

Este componente garantiza una separación clara de responsabilidades entre la lógica de negocio y la interfaz, promoviendo una arquitectura limpia y reactiva. Además, centraliza toda la interacción con la IA, lo que facilita la escalabilidad y el mantenimiento del sistema.

## Componente de Interfaz: MainIA

MainIA es la actividad principal de la aplicación responsable de inicializar la interfaz de usuario para la interacción con el modelo de inteligencia artificial **Gemini**. Esta actividad utiliza **Jetpack Compose**, el moderno sistema de UI declarativa de Android, para renderizar componentes de manera eficiente y responsive.

## Función: onCreate()

Dentro del ciclo de vida de la actividad, el método `onCreate()` realiza las siguientes acciones:

1. Configura el tema visual global de la aplicación mediante la función `Gemini2Theme`, asegurando una estética coherente y adaptada a dispositivos modernos.
  2. Establece una `Surface` de nivel raíz que actúa como contenedor principal del contenido visual.
  3. Lanza la pantalla principal de interacción con la IA, instanciando el componente `BakingScreen()`, el cual permite al usuario enviar texto e imágenes al modelo y recibir una respuesta generada.

## Importancia

MainIA representa el punto de entrada lógico y visual de la aplicación. Su diseño basado en Jetpack Compose proporciona ventajas como:

- Interfaz de usuario declarativa y reactiva.
  - Compatibilidad con el patrón MVVM.
  - Integración fluida con temas, estados y navegaciones modernas.

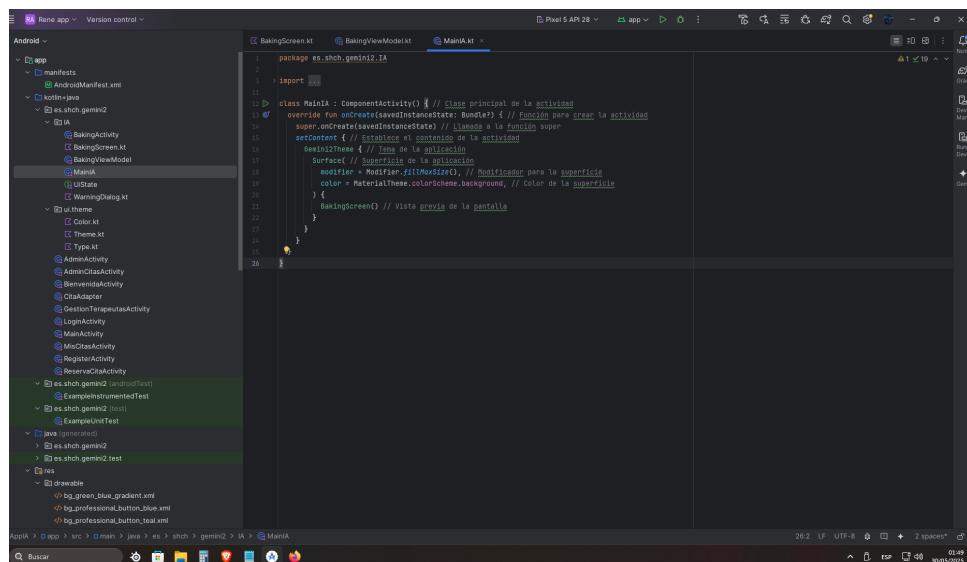


Figura 4.19: Pantalla de consulta médica con IA

Esta actividad cumple un rol fundamental al orquestar la carga inicial de la aplicación y facilitar el acceso inmediato del usuario al sistema de recomendaciones clínicas mediante inteligencia artificial.

## Modelo de Estado: UiState

**UiState** es una interfaz *sellada (sealed interface)* que representa los diferentes estados posibles en la interacción del usuario con el modelo de inteligencia artificial **Gemini**. Este patrón permite manejar la interfaz de usuario de manera reactiva y estructurada, proporcionando una forma clara de responder a los cambios de estado durante la ejecución.

### Estados definidos

- **Initial:** Estado predeterminado antes de que ocurra cualquier interacción. Representa una interfaz limpia o en espera.
- **Loading:** Indica que la solicitud del usuario está siendo procesada por el modelo de IA. Generalmente se asocia con un indicador visual de carga.
- **Success:** Representa una ejecución exitosa. Contiene la respuesta generada por la IA en forma de texto, lista para ser mostrada en la interfaz.
- **Error:** Indica que ha ocurrido un fallo durante el proceso (ya sea por conectividad, datos inválidos o error del modelo). Incluye un mensaje descriptivo para informar al usuario.

### Importancia

El uso de una interfaz sellada como **UiState** facilita:

- El control centralizado de la lógica de estado.
- La implementación de interfaces reactivas en **Jetpack Compose**.
- Una mayor claridad y mantenibilidad del código.

```

package es.sch.gemini2.IA
/*
 * Describe el estado de la generación de texto
 */
sealed interface Uistate {
    /**
     * Estado vacío al mostrar la pantalla por primera vez.
     */
    object Initial : Uistate

    /**
     * Aún cargando.
     */
    object Loading : Uistate

    /**
     * Se ha generado el texto.
     */
    data class Success(val outputText: String) : Uistate

    /**
     * Se produjo un error al generar el texto.
     */
    data class Error(val errorMessage: String) : Uistate
}

```

Figura 4.20: Código de UI

Cada cambio de estado se refleja directamente en la interfaz, garantizando una experiencia de usuario coherente y responsive durante la comunicación con la IA.

## Componente de Advertencia: AlertDialog

`AlertDialog` es un cuadro de diálogo diseñado para informar al usuario sobre las limitaciones del uso de la inteligencia artificial dentro de la aplicación. Su propósito es comunicar que el sistema de IA está destinado únicamente a consultas médicas básicas y no debe sustituir la atención profesional directa. Este componente se muestra de forma obligatoria antes de iniciar la interacción con el modelo generativo.

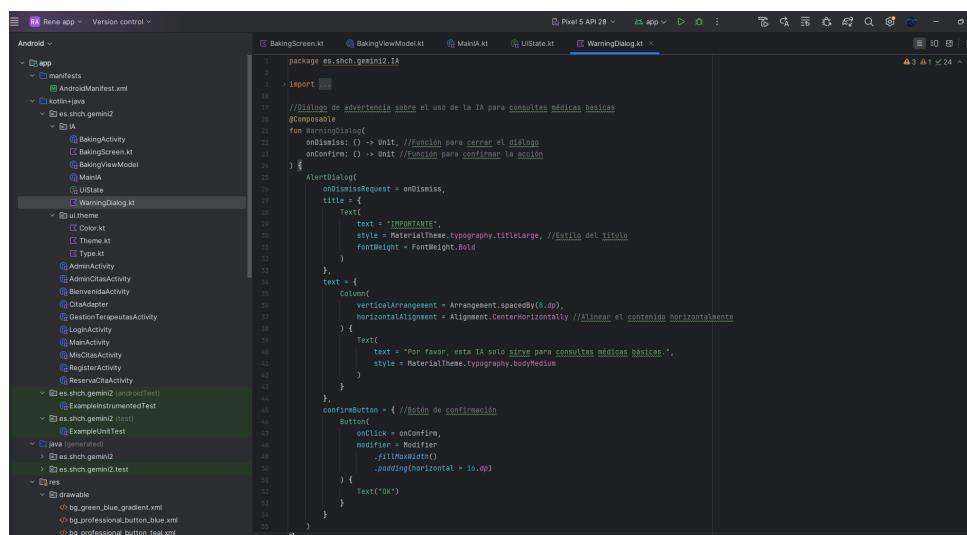
### Funciones de Callback

- `onDismiss`: Función que se ejecuta cuando el usuario cierra el cuadro de diálogo sin confirmar (por ejemplo, presionando fuera del área del diálogo o usando el botón de cerrar).
- `onConfirm`: Función que se activa al aceptar la advertencia, normalmente tras presionar el botón `OK`, permitiendo continuar hacia la interacción con la IA.

### Contenido del Diálogo

El componente está estructurado con los siguientes elementos visuales:

- **Título destacado:** "IMPORTANTE" — se presenta en la parte superior del diálogo con estilo resaltado.
- **Mensaje explicativo:** Un texto centrado que aclara que la IA está pensada únicamente para recomendaciones básicas, sin sustituir diagnósticos médicos profesionales.
- **Botón de confirmación:** Ocupa el ancho completo del cuadro de diálogo para mejorar la accesibilidad y claridad de la acción a ejecutar.



```

package es.schh.gemini2.IA
import ...

//Diálogo de advertencia sobre el uso de la IA para consultas médicas básicas
@Composable
fun WarningDialog(
    onDismiss: () -> Unit, //Función para cerrar el diálogo
    onConfirm: () -> Unit //Función para confirmar la acción
) {
    AlertDialog(
        onDismissRequest = onDismiss,
        title = {
            Text(
                text = "IMPORTANT",
                style = MaterialTheme.typography.titleLarge, //Estilo del título
                fontWeight = FontWeight.Bold
            )
        },
        text = {
            Column(
                verticalArrangement = Arrangement.spacedBy(8.dp),
                horizontalAlignment = Alignment.CenterHorizontally //Alinear el contenido horizontalmente
            ) {
                Text(
                    text = "Por favor, esta IA solo sirve para consultas médicas básicas.",
                    style = MaterialTheme.typography.bodyMedium
                )
            }
        },
        confirmButton = { //Botón de confirmación
            Button(
                onClick = onConfirm,
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(horizontal = 16.dp)
            ) {
                Text("OK")
            }
        }
    )
}

```

Figura 4.21: Pantalla de consulta médica con IA

## Importancia

Este diálogo cumple una función clave en términos de **ética y responsabilidad del uso de la IA**, ya que actúa como advertencia explícita antes de acceder a recomendaciones generadas automáticamente. Además, garantiza que el usuario tenga una comprensión adecuada del alcance del sistema y sus limitaciones.

## Presentación del Resultado Generado por la IA

Una vez que el usuario ha enviado un *prompt* textual, y el modelo de inteligencia artificial responde exitosamente, la aplicación actualiza su estado interno a `UiState.Success`.

Este estado contiene una variable denominada `outputText`, que representa el texto generado por el modelo Gemini 1.5 Flash. Dicho contenido corresponde a la respuesta de la IA ante la consulta del usuario, y puede incluir:

- Recomendaciones terapéuticas básicas.
- Interpretaciones médicas preliminares.
- Respuestas educativas relacionadas con el texto enviados.

## Visualización en la Interfaz

El resultado almacenado en `outputText` se muestra al usuario de forma clara, visible y accesible dentro de la pantalla `BakingScreen`, permitiendo una interpretación sencilla del contenido generado.



## Consulta IA



Escribe algo  
dame algunos tips para poder  
recuperar el movimiento de mi  
fractura de mano

Enviar

- ✓ **Elevación:** Mantén tu mano elevada por encima del nivel del corazón tanto como sea posible para reducir la hinchazón.
- ✓ **Ejercicios de rango de movimiento (ROM) pasivos:** Mientras tu mano esté inmovilizada, tu fisioterapeuta o un familiar puede ayudarte a mover suavemente tus dedos y muñeca dentro del rango de movimiento permitido. No fuerces el movimiento.
- ✓ **Compresas de hielo:** Aplica hielo durante 15-20 minutos varias veces al día para controlar la inflamación y el dolor. Nunca apliques hielo directamente sobre la piel; usa una toalla.
- ✓ **Fase de Rehabilitación (una vez que se retira el yeso o la férula):**
- ✓ **Ejercicios de rango de movimiento continuos:** Comienza a mover gradualmente tus

Figura 4.22: Pantalla de consulta médica con IA

### Citación Técnica

La integración se basa en el modelo Gemini, entrenado con auto-supervisión multimodal y ajuste final mediante aprendizaje supervisado y reforzado (G. DeepMind, (2023); Ouyang et al., (2022)).

## Arquitectura del funcionamiento de la IA

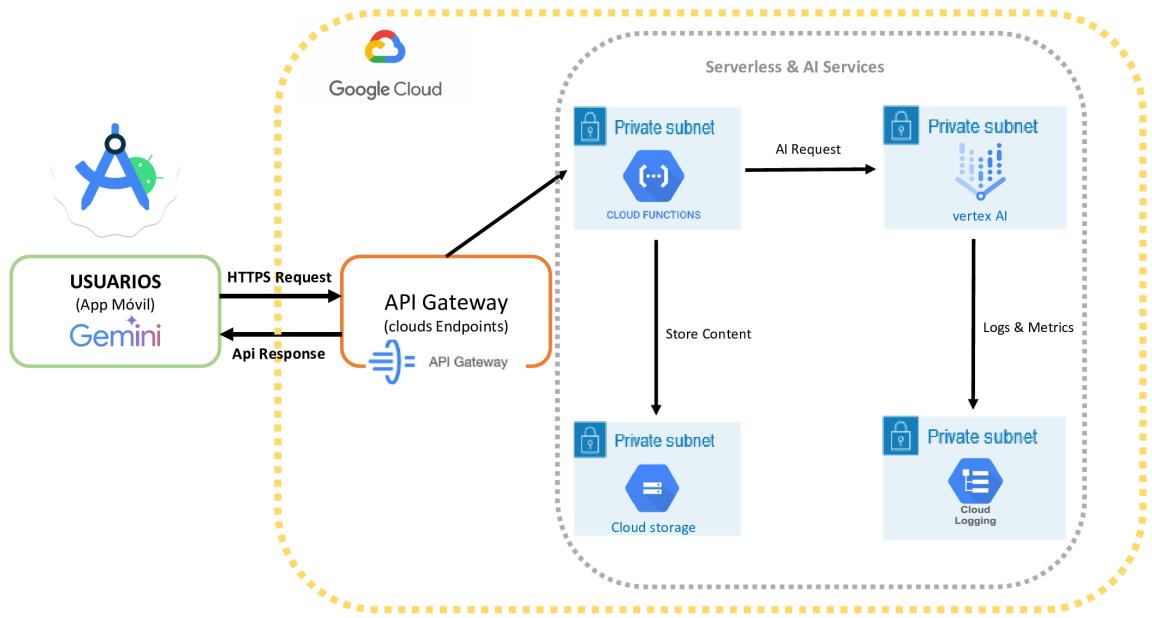
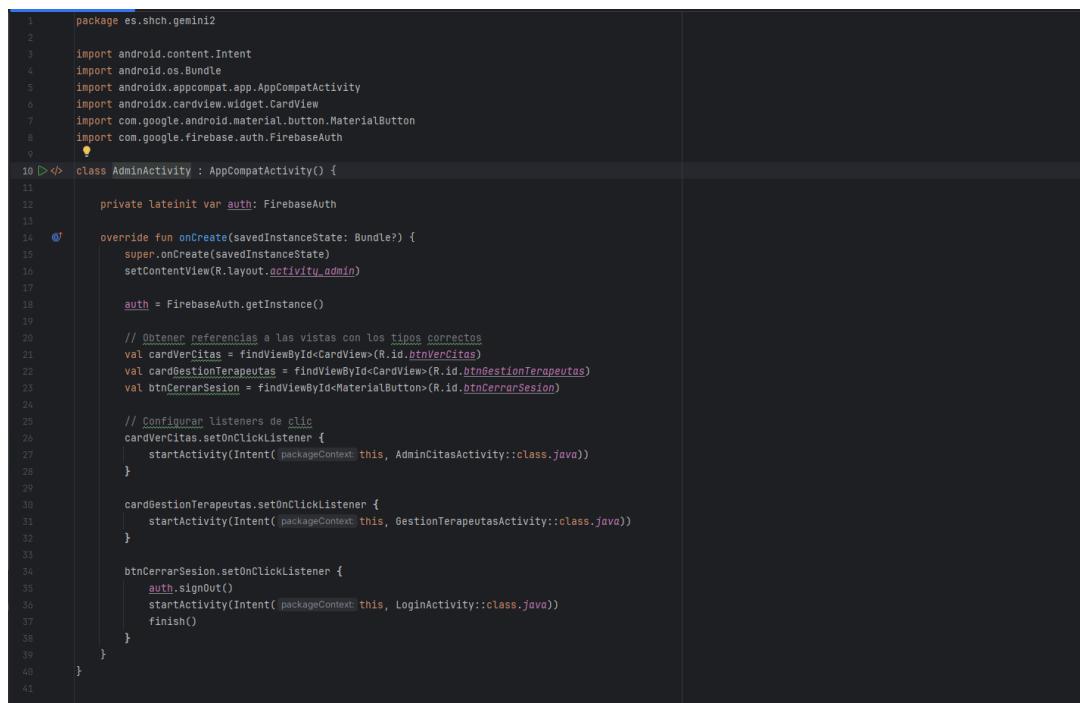


Figura 4.23: Nota: Elaboración Propia.

#### 4.3.2. Módulo 5: Panel de Administración

- **Acceso:** Exclusivo para administradores autenticados.
- **Características:**
  - Visualización de citas de todos los usuarios.
  - Asignación de médicos y horarios.
  - Confirmación por SMS utilizando la API de Twilio.

### Clases



```
1 package es.shch.gemini2
2
3 import android.content.Intent
4 import android.os.Bundle
5 import androidx.appcompat.app.AppCompatActivity
6 import androidx.cardview.widget.CardView
7 import com.google.android.material.button.MaterialButton
8 import com.google.firebase.auth.FirebaseAuth
9
10 class AdminActivity : AppCompatActivity() {
11
12     private lateinit var auth: FirebaseAuth
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_admin)
17
18         auth = FirebaseAuth.getInstance()
19
20         // Obtener referencias a las vistas con los tipos correctos
21         val cardVerCitas = findViewById<CardView>(R.id.cardVerCitas)
22         val cardGestionTerapeutas = findViewById<CardView>(R.id.cardGestionTerapeutas)
23         val btnCerrarSesion = findViewById<MaterialButton>(R.id.btnCloseSession)
24
25         // Configurar listeners de clic
26         cardVerCitas.setOnClickListener {
27             startActivity(Intent(packageContext, AdminCitasActivity::class.java))
28         }
29
30         cardGestionTerapeutas.setOnClickListener {
31             startActivity(Intent(packageContext, GestionTerapeutasActivity::class.java))
32         }
33
34         btnCerrarSesion.setOnClickListener {
35             auth.signOut()
36             startActivity(Intent(packageContext, LoginActivity::class.java))
37             finish()
38         }
39     }
40 }
41 }
```

Figura 4.24: Interfaz de AdminActivity

- **AdminActivity.kt:** Actividad principal para administradores que utiliza CardView como menú de navegación. Incluye funcionalidades para gestionar citas médicas, administrar terapeutas y cerrar sesión mediante Firebase Authentication.

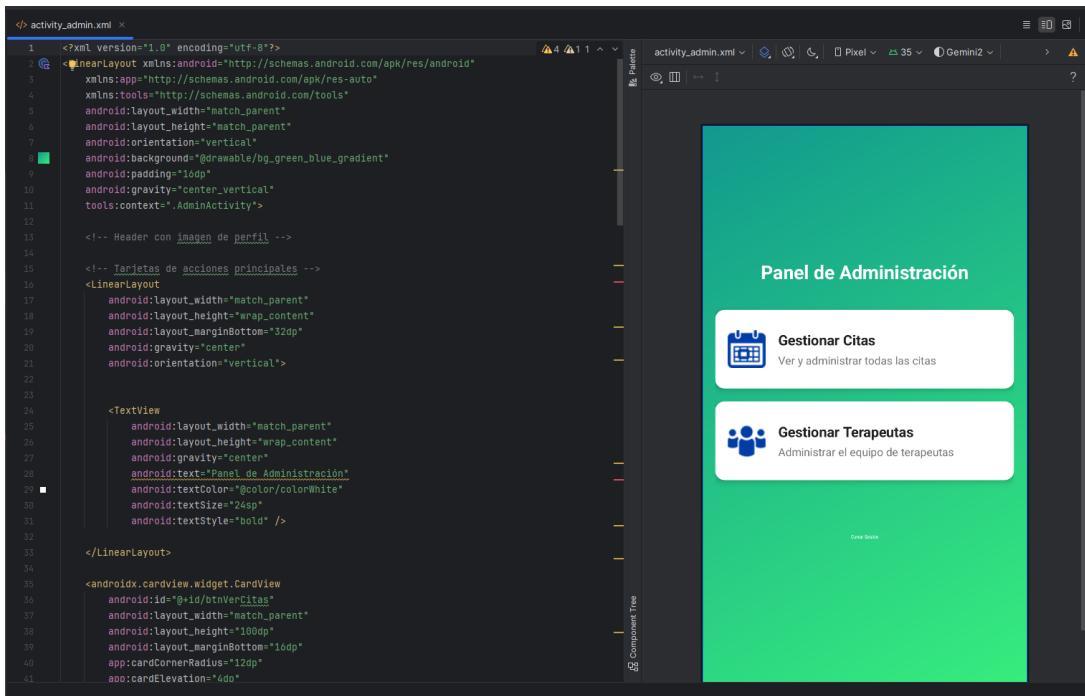


Figura 4.25: Interfaz de AdminCitasActivity

- **AdminCitasActivity.kt:** Módulo de administración de citas que se conecta con Firestore para recuperar y filtrar citas médicas. Permite búsquedas por nombre de paciente o número de DNI, mostrando los resultados en una vista de lista personalizada.

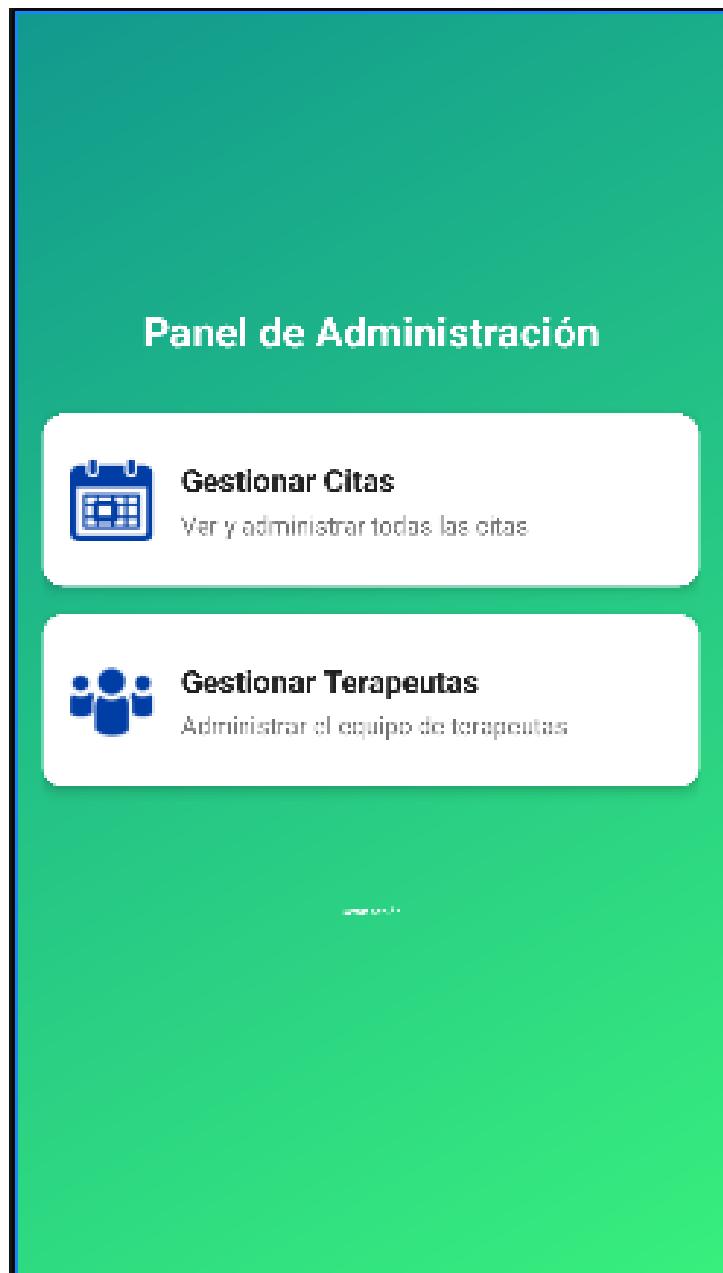


Figura 4.26: Panel de administración de citas

```

12     class GestionTerapeutasActivity : AppCompatActivity() {
13         override fun onCreate(savedInstanceState: Bundle?) {
14             setContentView(R.layout.activity_gestion_terapeutas)
15
16             db = FirebaseFirestore.getInstance()
17             listaTerapeutas = findViewById(R.id.lvTerapeutas)
18             etNombre = findViewById(R.id.etNombreTerapeuta)
19             etEspecialidad = findViewById(R.id.etEspecialidad)
20             btnAgregar = findViewById(R.id.btnAddTerapeuta)
21
22             cargarTerapeutas()
23
24             btnAgregar.setOnClickListener {
25                 agregarTerapeuta()
26             }
27         }
28
29
30         private fun cargarTerapeutas() {
31             db.collection( collectionPath: "terapeutas").get()
32                 .addOnSuccessListener { documents ->
33                     val terapeutas = mutableListOf<String>()
34                     for (document in documents) {
35                         val nombre = document.getString( field: "nombre") ?: ""
36                         val especialidad = document.getString( field: "especialidad") ?: ""
37                         terapeutas.add("$nombre - $especialidad")
38                     }
39
40                     val adapter = ArrayAdapter( context: this, android.R.layout.simple_list_item_1, terapeutas)
41                     listaTerapeutas.adapter = adapter
42                 }
43             }
44
45
46         private fun agregarTerapeuta() {
47             val nombre = etNombre.text.toString().trim()
48             val especialidad = etEspecialidad.text.toString().trim()
49
50             if (nombre.isNotEmpty() && especialidad.isNotEmpty()) {
51                 val terapeuta = hashMapOf(
52                     "nombre" to nombre,
53                     "especialidad" to especialidad
54                 )
55             }
56         }
57     }
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
707
708
709
709
710
711
712
713
714
715
715
716
717
717
718
719
719
720
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
163
```

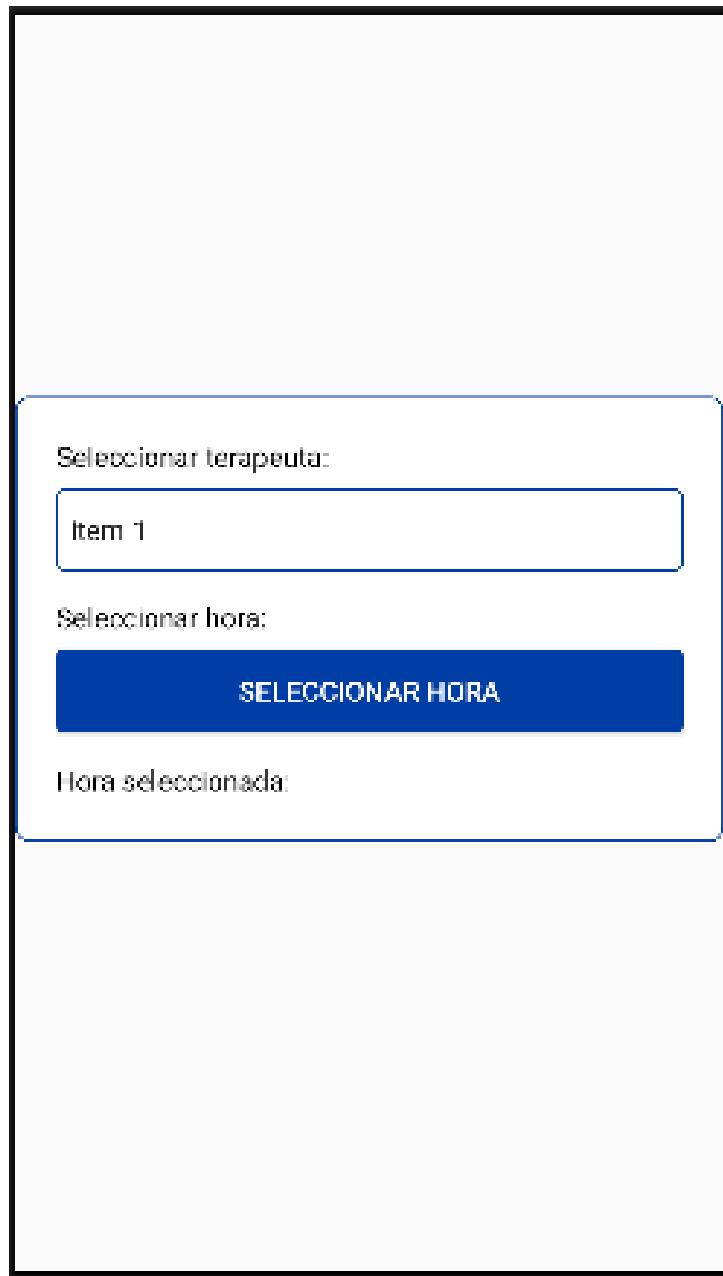


Figura 4.28: Pantalla para asignar cita

### 4.3.3. Arquitectura del sistema

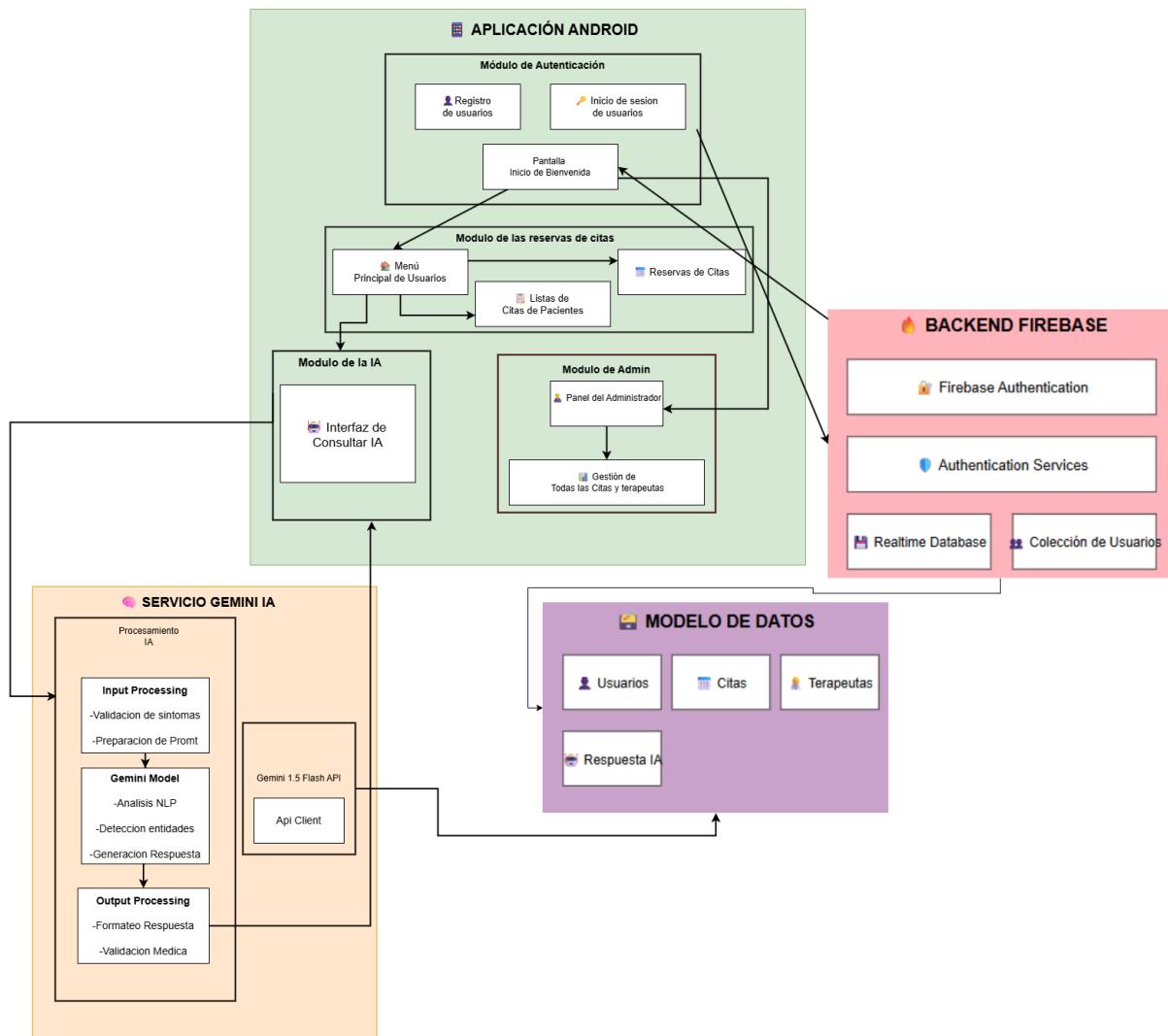


Figura 4.29: Nota: Elaboración propia

#### 4.3.4. Cronograma de ejecución del Proyecto

Nº	Tarea	Inicio	Duración	Finalización	Nombre recursos	% Completado	Días completado
1	Sistema de Cotización	18-feb	20	10-mar		92%	18,40
2	Gestión de proyecto	20-feb	20	12-mar	Fabian Coras	92%	18,40
3	+Inicio	25-feb	20	17-mar	Fabian Coras	100%	20,00
6	- Planificación e investigación	25-feb	20	17-mar	Sebastian Huertas	100%	20,00
4	Definición del Objetivo del proyecto	27-feb	5	4-mar	Fabian Coras	100%	5,00
5	Realizar especificación de funcionalidad	4-mar	5	9-mar	Fabian Coras	100%	5,00
7	Recopilar requisitos	4-mar	5	9-mar	Sebastian Huertas	100%	5,00
8	Elaborar una linea de alcance	4-mar	5	9-mar	Kevin Cari	100%	5,00
3	+Elaboración	6-mar	13	19-mar	Fabian Coras	90%	11,70
6	- Requerimientos	6-mar	10	16-mar	Sebastian Huertas	100%	10,00
10	Analisis de requerimiento	8-mar	6	14-mar	Sebastian Huertas	90%	5,40
8	Matriz de Requerimiento	8-mar	4	12-mar	Sebastian Huertas	90%	3,60
6	- Diseño y Analisis del Software	13-mar	3	16-mar	Fabian Coras	100%	3,00
8	Caso de Uso	13-mar	3	16-mar	Kevin Cari	100%	3,00
3	+Construcción	18-mar	51	8-may	Fabian Coras	100%	51,00
9	- Diseño e implementación	18-mar	43	30-abr	Fabian Coras	100%	43,00
11	Diseño en Android studio	20-mar	5	25-mar	Fabian Coras	100%	5,00
12	Creacion de clases en kotlin android studio	20-mar	10	30-mar	Fabian Coras	100%	10,00
13	Creación de login y conexión BD	20-mar	4	24-mar	Fabian Coras	100%	4,00
14	Creacion de registro y conexión BD	20-mar	4	24-mar	Fabian Coras	100%	4,00
15	Creacion de las reservas de cita	3-abr	2	5-abr	Sebastian Huertas	100%	2,00
16	Creacion del diseño admin	10-abr	2	12-abr	Fabian Coras	100%	2,00
17	Implementación de la IA	24-abr	5	29-abr	Sebastian Huertas	100%	5,00
18	- Creación de reportes	27-abr	3	30-abr	Kevin Cari	100%	3,00
19	Creación de reportes cotización en PDF	27-abr	3	30-abr	Kevin Cari	100%	3,00
20	Creacion de reportes de boletas y facturación	27-abr	5	2-may	Kevin Cari	100%	5,00
21	+Transición	29-abr	8	7-may		85%	6,80
22	- Pruebas de Crud y formularios	3-may	3	6-may	Kevin Cari	80%	2,40
23	Curd de Prueba	8-may	5	13-may	Sebastian Huertas	90%	4,50
3	Pruebas de formulas en formulario cotización	10-may	9	19-may	Sebastian Huertas	75%	6,75
24	Pruebas de Inserción de artículos según su pro	13-may	9	22-may	Kevin Cari	75%	6,75
25	Pruba de validar registros de artículos	23-may	3	26-may	Sebastian Huertas	90%	2,70
26	+ Aprobar el proyecto y dar visto bueno	27-may	6	2-jun	Jose Tiznado	60%	3,60

Figura 4.30: Actividades del proyecto

#### 4.3.5. Diagrama de Proceso Mejorado

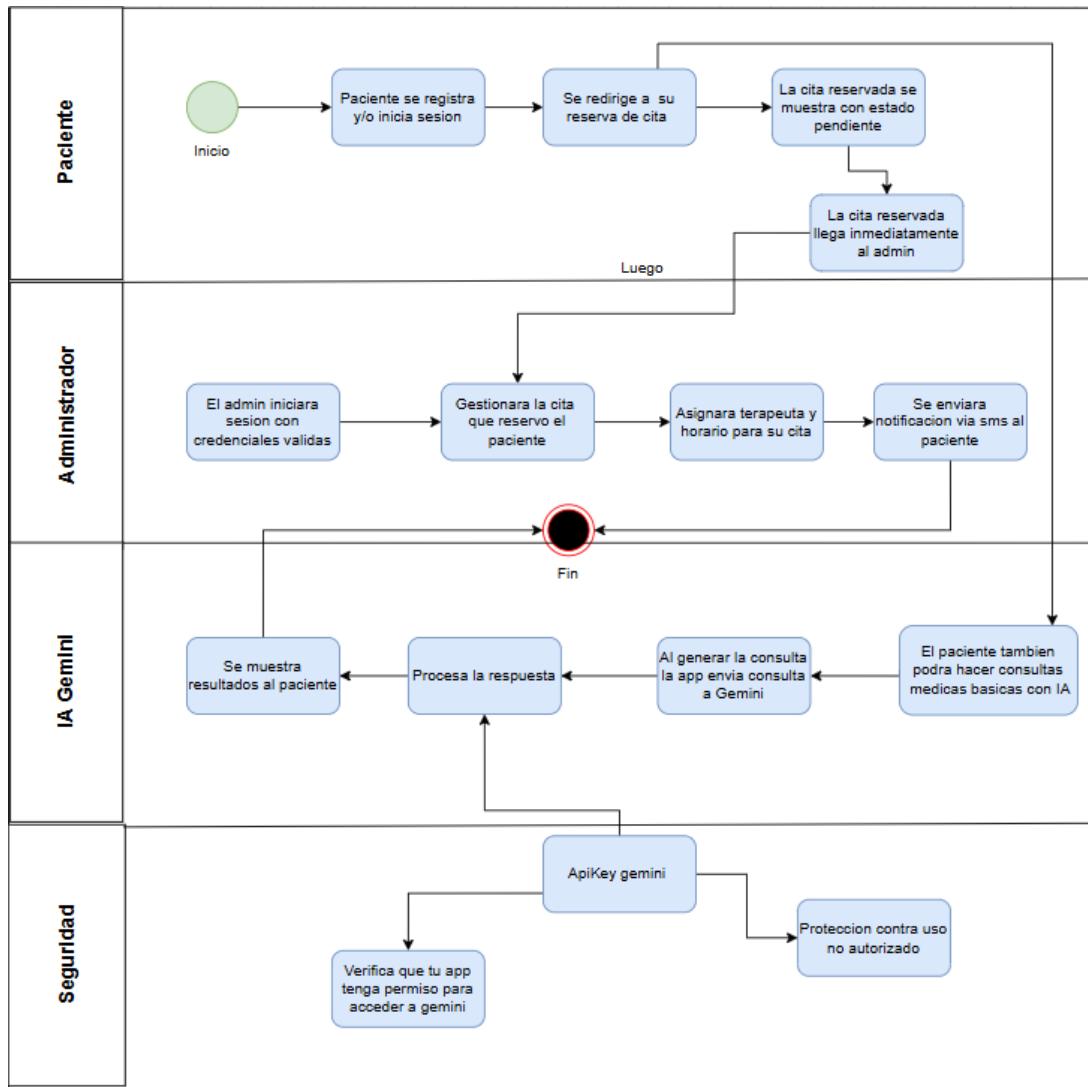


Figura 4.31: Diagrama de mejora



Figura 4.32: Diagrama de Gantt

# Capítulo 5

## COSTO DE IMPLEMENTACIÓN DE LA MEJORA / INNOVACIÓN / CREATIVIDAD

### 5.1. Costo de materiales

Se procede a describir el costo de materiales utilizado en el desarrollo del sistema.

Servidor	País	Nombre	Costo (S/)	Costo (US\$)
Dominio	Perú	—	S/00.00	\$00.00
		<b>Total</b>	<b>S/00.00</b>	<b>\$00.00</b>

Cuadro 5.1: *Costos estimados de materiales utilizados para la implementación del proyecto.*

### 5.2. Costo de mano de obra

Actividad	Cantidad	Días laborables	Costo unitario	Total (S/)	Total (US\$)
Desarrollador	1	105	S/00.00	S/00.00	\$00.00
Desarrollador	1	105	S/00.00	S/00.00	\$00.00
Desarrollador	1	105	S/00.00	S/00.00	\$00.00
<b>Total</b>				<b>S/00.00</b>	<b>\$00.00</b>

Cuadro 5.2: *Tabla de costos de mano de obra estimados para el desarrollo del proyecto.*

## 5.3. Costo de maquinaria, herramientas y equipos

### 5.3.1. Costo de depreciación de herramientas

Características de soporte de equipos

Nombre	Descripción
Características PC	Intel Core i7 – Disco Duro Sólido 1TB
Valor inicial	S/3,773.50
Vida útil en años	4 años
Valor residual	S/377.35

Cuadro 5.3: *Detalle técnico y económico del equipo Laptop 01 utilizado en el desarrollo del proyecto.*

## 5.4. Depreciación de las computadoras

Se define que el costo de depreciación de la herramienta de la PC según el cálculo por año es de S/800.00, que se estima en la durabilidad que tendrá la PC.

Cuadro 5.4: *Cálculo de depreciación anual*

Nombre	Cálculo
Depreciación	$S/ 800.00 = S/ 4000.00 / 5 \text{ años}$

Año	Valor anterior (S/)	Depreciación anual (S/)	Valor después de depreciar (S/)
1	S/ 4,500.00	S/ 800.00	S/ 3,700.00
2	S/ 3,700.00	S/ 800.00	S/ 2,900.00
3	S/ 2,900.00	S/ 800.00	S/ 2,100.00
4	S/ 2,100.00	S/ 800.00	S/ 1,300.00
5	S/ 1,300.00	S/ 800.00	S/ 500.00

Cuadro 5.5: *Cálculo de depreciación anual del equipo durante su vida útil estimada de 5 años.*

## 5.5. Otros costos de implementación de la mejora

### 5.5.1. Costo de Servicios y Consumo de Luz

El costo de servicio de luz se calculó por el consumo de luz y la estimación de los días trabajados por los desarrolladores. Se estimó una cantidad de 120 días laborados por 8 horas al día, generando un total de consumo mensual con una tarifa de S/ 0.85 por kWh.

<b>Nombre</b>	<b>Total</b>
Consumo W	180 W
Convertir Potencia	180 W / 1000 = 0.18 kW
Días	30 días
Horas	8 horas
Cálculo - Mes	8 horas × 30 días = 240 h/mes
Cálculo KWh	0.18 kW × 240 h = 43.2 kWh
Tarifa Precio	S/ 0.8039
Total Mes	S/ 0.8039 × 43.2 kWh = S/ 34.74
Cálculo x Día	S/ 34.74 / 30 días = S/ 1.16/día
Días x Programador	105 días
Total Luz	105 días × S/ 1.16 = S/ 121.77

Cuadro 5.6: *Lista de gastos personales del consumo de servicio de luz e internet*

### 5.5.2. Lista de gastos personales

A continuación, se muestran los costos adicionales en el desarrollo del proyecto detallando los costos de luz y otros servicios adicionales:

<b>Descripción</b>	<b>Cant.</b>	<b>Precio Unitario</b>	<b>Total (S/)</b>	<b>Total (USD)</b>
Servicio de Luz x Programadores	1	S/ 121.77	S/ 121.77	\$31.96
Internet	1	S/ 160.00	S/ 160.00	\$41.99
<b>Total</b>			<b>S/ 281.77</b>	<b>\$73.95</b>

Cuadro 5.7: *Costos adicionales de luz e internet*

### 5.5.3. Costo total de la implementación de la mejora

<b>Descripción</b>	<b>Total (S/)</b>	<b>Dólar (USD)</b>
Costo de Material	S/ 0.00	\$0.00
Costo de Mano de Obra	S/ 0.00	\$0.00
Costo Maquinaria, herramientas y equipos	S/ 800.00	\$209.83
Otros Costos	S/ 281.77	\$73.93
<b>Total</b>	<b>S/ 1,081.77</b>	<b>\$283.76</b>

Cuadro 5.8: *Total de la implementación del proyecto*

# Capítulo 6

## EVALUACIÓN TÉCNICAS Y ECONÓMICAS DE LA MEJORA / INNOVACIÓN / CREATIVIDAD

### 6.1. Beneficio técnico y / o económico esperado de la mejora / innovación / creatividad

Con la implementación del aplicativo móvil inteligente para la Clínica René Rehabilitación S.A.C., se optimizaron los siguientes procesos y se generaron beneficios tanto técnicos como operativos para el centro de salud:

- **Agilización en la gestión de citas médicas:** Permitió a los pacientes reservar, visualizar y organizar sus citas desde el aplicativo, sin necesidad de acudir físicamente.
- **Automatización de recomendaciones de salud:** Mediante un chatbot con inteligencia artificial (Gemini), que brinda orientación básica e inmediata a los pacientes.
- **Reducción de la carga administrativa:** Al contar con un panel de gestión que permite al personal asignar horarios, terapeutas y confirmar citas vía SMS.
- **Mejora en la calidad del servicio al paciente:** Ofreciendo una atención más personalizada, rápida y eficiente.
- **Modernización tecnológica del centro:** Al incorporar herramientas innovadoras de IA y móviles, posicionando a la clínica como una institución vanguardista en salud.

#### 6.1.1. Relación Beneficio / Costo

El beneficio obtenido del presente proyecto radica en la optimización de los procesos administrativos y de atención médica mediante la digitalización y automatización inteligente:

- Reducción del tiempo y errores en la asignación de citas médicas al eliminar procesos manuales.

- Ahorro en recursos administrativos y operativos, ya que el sistema automatiza funciones que antes requerían intervención constante.
- Disminución de costos de atención inicial, gracias a las respuestas preliminares del chatbot con IA, que filtra consultas simples.
- Incremento en la satisfacción del paciente, lo cual impacta positivamente en la fidelización y la reputación de la clínica.
- Escalabilidad del sistema para futuras mejoras, permitiendo integrar nuevos módulos sin rehacer el sistema base.

## 6.2. Identificación de pérdida económica

A continuación, se identifican las causas que generaron pérdida económica en el proceso, las cuales son consecuencia directa de la falta de un sistema automatizado y del bajo compromiso del paciente en los ejercicios domiciliarios.

### Causa 1: Falta de herramientas automatizadas para seguimiento remoto

Cuadro 6.1: Pérdidas económicas por falta de herramientas automatizadas para seguimiento remoto

Causa	Evento	Pérdida mensual (S/)	Pérdida anual (S/)
Falta de herramientas automatizadas para seguimiento remoto	Falta de retroalimentación en tiempo real	S/ 100.00	S/ 1,200.00
Falta de herramientas automatizadas para seguimiento remoto	Demora en identificar retrocesos en la recuperación	S/ 90.00	S/ 1,080.00
Falta de herramientas automatizadas para seguimiento remoto	Interrupciones en el tratamiento por falta de monitoreo	S/ 90.00	S/ 1,080.00
<b>Total</b>			<b>S/ 3,360.00</b>

### Causa 2: Baja adherencia a ejercicios en casa

Cuadro 6.2: Pérdidas económicas por baja adherencia a ejercicios en casa

Causa	Evento	Pérdida mensual (S/)	Pérdida anual (S/)
Baja adherencia a ejercicios en casa	Olvido frecuente de los ejercicios asignados	S/ 120.00	S/ 1,440.00
Baja adherencia a ejercicios en casa	Desmotivación por falta de seguimiento	S/ 130.00	S/ 1,560.00
Baja adherencia a ejercicios en casa	Abandono del tratamiento	S/ 100.00	S/ 1,200.00
<b>Total</b>			<b>S/ 4,200.00</b>

### 6.2.1. Total promedio de pérdidas anuales

Cuadro 6.3: Total promedio de pérdidas anuales

Causa	Descripción	Anual (S/)
C1	Falta de herramientas automatizadas para el seguimiento remoto	S/ 3,360.00
C5	Baja adherencia a ejercicios en casa	S/ 4,200.00
<b>Total promedio de pérdida</b>		<b>S/ 7,560.00</b>

### 6.2.2. Total de la implementación del proyecto

Cuadro 6.4: Costo y eficiencia de la implementación del proyecto

Pérdida anual (S/)	Representa (%)	Costo total de la implementación (S/)	% de eficiencia de recuperación
S/ 7,560.00	100 %	S/ 364.00	90 %

### 6.2.3. Total promedio de pérdidas anuales después de la mejora

Cuadro 6.5: Comparación de pérdidas antes y después de la mejora

Descripción	Monto (S/)
Pérdidas anuales	S/ 7,560.00
Pérdidas después de la mejora	S/ 756.00
<b>Total ahorro</b>	<b>S/ 6,804.00</b>

#### 6.2.4. Relación beneficio/costo

Cuadro 6.6: Relación beneficio/costo

Concepto	Monto (S/)
Beneficio	S/ 6,804.00
Costo de inversión	S/ 1,081.77

**Análisis:** Por cada sol invertido se recupera S/ 6.29.

#### 6.2.5. Tiempo de recuperación de la inversión

Cuadro 6.7: Comparación de pérdidas antes y después de la mejora

Descripción	Monto (S/)
Pérdidas anuales	S/ 7,560.00
Pérdidas después de la mejora	S/ 756.00
<b>Total ahorro</b>	<b>S/ 6,804.00</b>

$$0.159 \text{ años} \times 12 \text{ meses} = 1.91 \text{ meses}$$

$$0.91 \text{ meses} \approx 27.24 \text{ días}$$

**Análisis:** Significa que en 1 mes y 27 días se recupera lo invertido.

# Capítulo 7

## Conclusiones respecto a los objetivos del Proyecto de Mejora/Innovación/Creatividad

### 7.1. Resultados

La implementación del aplicativo móvil con inteligencia artificial en la Clínica RENE Reeducación Neuromotora S.A.C. ha representado un cambio significativo en la forma en que se realiza el seguimiento y la gestión de las citas médicas y del estado del paciente fuera del entorno clínico. Antes del desarrollo de esta solución, la institución enfrentaba una **problemática importante: la falta de un control continuo y personalizado una vez que los pacientes dejaban la clínica**. Esto generaba **una brecha considerable en el proceso terapéutico**, pues los pacientes **no siempre seguían correctamente las indicaciones, y los terapeutas debían basarse en relatos subjetivos para evaluar el avance**, lo que **comprometía la calidad del tratamiento**.

Con el nuevo sistema, esta situación ha cambiado notablemente. Ahora, los pacientes pueden agendar, modificar y cancelar sus citas desde la aplicación móvil, sin necesidad de acudir físicamente o hacer llamadas. Esto ha reducido los tiempos de espera y ha permitido una mejor organización de la atención médica, beneficiando tanto a los usuarios como al personal administrativo.

Además, la integración del sistema de inteligencia artificial, mediante la API de Gemini, ha permitido brindar recomendaciones básicas de salud personalizadas, en tiempo real, directamente desde el teléfono del paciente. Ante molestias comunes como fatiga, rigidez muscular o dolor leve, el usuario recibe orientación inmediata y sencilla, lo que fortalece su autonomía y continuidad en el tratamiento. **Esta funcionalidad ha suplido la falta de guía entre sesiones presenciales**, ofreciendo una herramienta útil para el autocuidado en casa.

Asimismo, la carga operativa del personal se ha reducido significativamente, ya que muchas tareas anteriores manuales, como el registro de citas o la confirmación de horarios, ahora son gestionadas automáticamente por el sistema. La experiencia del paciente también ha mejorado al contar con recordatorios y confirmaciones por notificación, evitando olvidos y descoordinaciones.

En resumen, la aplicación ha permitido **cerrar la brecha entre el tratamiento clínico y el seguimiento domiciliario**, ofreciendo un canal confiable, eficiente y automatizado de apoyo terapéutico. Con ello, se ha logrado no solo una mejor administración de citas, sino también una continuidad más sólida y personalizada del proceso de recuperación.

## Discusión

La implementación del sistema inteligente de orientación en salud terapéutica ha evidenciado su potencial como herramienta de apoyo en procesos de atención clínica. En particular, ha demostrado ser eficaz en la generación de recomendaciones personalizadas a partir de síntomas ingresados por el usuario, gracias al uso del modelo de lenguaje generativo Gemini 1.5 Flash.

Este enfoque contrasta con la propuesta presentada por Yamao y Ernesto, (2020), quien desarrolló una aplicación de recomendaciones alimentarias enfocada en reducir la anemia infantil en un contexto escolar. Aunque ambas investigaciones comparten el objetivo general de ofrecer recomendaciones en el ámbito de la salud mediante tecnologías móviles, se diferencian en diversos aspectos: la población objetivo (niños vs. pacientes en rehabilitación), el tipo de intervención (nutricional vs. terapéutica) y el modelo tecnológico empleado (reglas predefinidas vs. IA generativa).

A su vez, investigaciones como la de Liu, (2022) introducen modelos de lenguaje para ofrecer orientación médica automatizada mediante el uso de \*transformers\*. En dicho estudio, se empleó un modelo de lenguaje para responder preguntas médicas generales, obteniendo niveles de precisión cercanos al 75 % en consultas comunes. Si bien su foco estuvo en la consulta médica general, nuestro sistema va un paso más allá al integrar Gemini 1.5 Flash, un modelo multimodal que permite una mayor comprensión contextual y respuestas más detalladas en lenguaje natural.

Otro estudio comparable es el de Hosseini y Rahmani, (2021), donde se desarrolló un sistema de recomendación terapéutica basado en técnicas de minería de datos sobre historiales clínicos. A diferencia de nuestro enfoque, que utiliza lenguaje natural para interactuar con el usuario, dicho sistema requería el análisis estructurado de grandes volúmenes de datos previos, lo que limitaba su aplicabilidad en contextos de primer contacto o sin historial médico digitalizado.

Mientras que sistemas previos como los mencionados operan sobre bases cerradas y modelos deterministas o entrenados localmente, el sistema aquí propuesto incorpora procesamiento de lenguaje natural en tiempo real, utilizando el modelo generativo Gemini 1.5 Flash a través de Vertex AI. Este modelo alcanza una precisión factual del 79 % en tareas de extracción de información Anakin, (2024), lo que permite una interpretación más precisa y contextual de los síntomas descritos por los usuarios.

Este avance técnico amplía considerablemente el alcance del sistema, haciendo posible su uso en diversos escenarios clínicos más allá de los casos contemplados originalmente. A diferencia de enfoques anteriores más estáticos, la flexibilidad del modelo permite adap-

tarse a distintos perfiles y necesidades terapéuticas, brindando una herramienta adaptable a distintos tipos de pacientes y consultas.

Finalmente, desde el punto de vista ético y funcional, se han establecido límites claros: las recomendaciones emitidas no sustituyen una consulta médica, sino que actúan como apoyo previo a una interacción presencial con un especialista. Este mecanismo de cierre mitiga el riesgo de una dependencia indebida a la tecnología, garantizando que el sistema actúe como complemento y no como reemplazo del juicio clínico humano.

### **7.1.1. Conclusión**

En conclusión, el desarrollo del sistema de inteligencia artificial para la optimización de citas médicas y gestión de pacientes en la Clínica RENE S.A.C. ha demostrado ser una solución innovadora, eficaz y adaptable a las necesidades reales del entorno médico peruano. No solo se han alcanzado los objetivos establecidos al inicio del proyecto, sino que se ha superado la expectativa en cuanto a precisión, eficiencia operativa y experiencia del usuario. Este trabajo representa un paso importante hacia la modernización de los servicios de salud mediante tecnologías inteligentes, sentando las bases para futuras ampliaciones del sistema, como integración con historiales clínicos electrónicos, alertas de emergencia, o aplicaciones móviles orientadas al paciente.

# Capítulo 8

## Recomendaciones para la empresa respecto del Proyecto de Mejora/Innovación/Creatividad

### 8.1. Recomendaciones

- **Implementación de un Sistema de Monitoreo Continuo:** Se desarrollarán e integrarán herramientas tecnológicas que permitan el monitoreo remoto del progreso de los pacientes mediante sensores biométricos o dispositivos *wearables*. Esto facilitará la recolección de datos objetivos en tiempo real y mejorará la precisión del seguimiento terapéutico.
- **Personalización de Recomendaciones con IA:** Se utilizarán algoritmos de inteligencia artificial para analizar los datos recopilados y generar recomendaciones terapéuticas adaptadas a las necesidades individuales de cada paciente, incluyendo ajustes en ejercicios, intensidad y frecuencia según su evolución.
- **Notificaciones Automáticas y Alertas:** Se implementará un sistema de alertas que notifique a terapeutas y pacientes sobre retrocesos, incumplimientos o cambios significativos en el estado de salud, permitiendo intervenciones oportunas y mejorando la adherencia al tratamiento.
- **Capacitación del Personal y Pacientes:** Se realizarán talleres de capacitación para el personal médico y los pacientes sobre el uso de la aplicación móvil y las nuevas tecnologías integradas, asegurando una adopción efectiva y maximizando los beneficios del sistema.
- **Evaluación Continua y Mejoras Iterativas:** Se establecerá un protocolo de evaluación periódica del sistema para identificar áreas de mejora y actualizar funcionalidades según comentarios de los usuarios y avances tecnológicos.
- **Garantía de Seguridad y Privacidad:** Se asegurará el cumplimiento de las normativas de protección de datos Congreso de la República del Perú, (2021) mediante medidas de encriptación y control de acceso para proteger la información sensible de los pacientes.

- **Escalabilidad e Integración con Otros Sistemas:** La solución se diseñará con capacidad para integrarse con otros sistemas de salud electrónicos (como historiales clínicos digitales) y será escalable para atender a más pacientes sin comprometer el rendimiento.
- **Enfoque en la Experiencia del Usuario:** Se optimizará la interfaz de la aplicación móvil para que sea intuitiva, accesible y compatible con dispositivos de diferentes gamas, garantizando una experiencia fluida para todos los usuarios.

# Referencias

- Alammar, J. ((2018)). The illustrated transformer. <https://jalammar.github.io/illustrated-transformer/>
- Anakin. ((2024)). Gemini 1.5 Flash: Google's High-Speed AI Model. <https://anakin0.blogspot.com/2024/05/gemini-15-flash-googles-high-speed-ai.html>
- Android Activity Team. ((2025)). androidx.activity.compose [Accedido el 2 de junio de 2025].
- Android Lifecycle Team. ((2025)). androidx.lifecycle.viewmodel.compose [Accedido el 2 de junio de 2025].
- Android UI Team. ((2025)). Layouts | Android Developers [Accedido el 2 de junio de 2025].
- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., ... Wu, Y. ((2023), mayo). PALM 2 Technical Report. <https://arxiv.org/abs/2305.10403>
- Arturo, S. C. J. ((2022), agosto). Sistema de gestión del plan nutricional basado en modelos de decisión para pacientes con enfermedades causadas por mala alimentación. <https://repositorioacademico.upc.edu.pe/handle/10757/661211>
- Augusto, V. S. J. ((2016)). Modelo de aprendizaje para sistemas de recomendación, caso: Curso Programación Web. <https://repositorio.unsa.edu.pe/items/90562efe-2ba9-4051-8c24-77b8bd844c98>
- Bendezu Castilla, R. P., & Ysla Parra, R. A. A. ((2020)). App de recomendaciones alimentarias para reducir la mala alimentación en casos de anemia en niños del colegio "Apóstol de Punchauca". <https://repositorio.usmp.edu.pe/handle/20.500.12727/6824>
- Brown, T. B., et al. ((2020)). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.
- Celia, F. A. ((2022)). Inteligencia artificial para “garantizar una vida sana y promover el bienestar”. Recomendaciones de IA confiable para mejorar el sistema de salud en la República De Chile en el marco del ODS 3 | Archivo Digital UPM. <https://oa.upm.es/71956/>
- Chatterjee, P., Sharma, A., & Rana, R. ((2022)). Chatbots in Healthcare: A Review of Use Cases and Technology Adoption. *Journal of Medical Systems*, 46(3), 1-14. <https://doi.org/10.1007/s10916-022-01768-9>
- Chávez, M. ((2022)). Sincronización de notificaciones en sistemas móviles híbridos para salud. *Revista de Innovación en TICs*, 6(1), 55-64.
- Chi, J. P., & Nichols, E. ((2016)). Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4, 357-370. [https://doi.org/10.1162/tacl\{\\\_}a\{\\\_}00104](https://doi.org/10.1162/tacl\{\_}a\{\_}00104)

- Congreso de la República del Perú. ((2021)). Ley N.º 29733 - Ley de Protección de Datos Personales [Última actualización en 2025]. <https://www.gob.pe/institucion/pcm/normas-legales/243470-29733>
- De Croon, R., Van Houdt, L., Htun, N. N., Štiglic, G., Vanden Abeele, V., & Verbert, K. ((2021)). Health Recommender Systems: Systematic review. *Journal of Medical Internet Research*, 23(6), e18035. <https://doi.org/10.2196/18035>
- DeepMind. ((2024)). Gemini 1.5: Technical Report [Accedido el 29 de mayo de 2025].
- DeepMind, G. ((2023)). *Gemini: A family of highly capable multimodal models* [Recuperado de DeepMind Blog]. <https://deepmind.google/discover/blogs/google-deepmind-unveils-gemini-1>
- Developers, A. ((2023)). Guide to app architecture: ViewModel [Consultado en junio de 2025]. <https://developer.android.com/topic/architecture>
- Developers, G. ((2023)). Transactions and Batched Writes | Firestore Documentation. [%5Curl%7Bhttps://firebase.google.com/docs/firestore/manage-data/transactions%7D](https://firebase.google.com/docs/firestore/manage-data/transactions#7D)
- Firebase Auth Team. ((2025)). Firebase Authentication for Android [Accedido el 2 de junio de 2025].
- Firebase Firestore Team. ((2025)). Cloud Firestore for Android [Accedido el 2 de junio de 2025].
- Francisco, A. L. J. ((2020), octubre). Software de recomendación médica basado en modelo de datos orientado a grafos con NEO4J. <https://repositorio.ucsm.edu.pe/items/ae5a8f7-bc38-49a7-a970-b40c88bf687f>
- García, L., Ramírez, S., & Núñez, P. ((2020)). Diseño de arquitectura híbrida para apps médicas móviles con Firebase y Android. *Revista Peruana de Ingeniería de Software*, 12(1), 34-42.
- Giannina, M. L. L. ((2019), noviembre). Implementación de un sistema de recomendación de medicamentos en base a la naturaleza del paciente geriátrico. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/15402>
- Google Android Studio Team. ((2025)). Android Studio - Android Developers [Accedido el 2 de junio de 2025].
- Google Cloud. ((2024)). Gemini API for Text: Best Practices for LLM-Based Interaction [Accedido el 29 de mayo de 2025].
- Google Identity Services. ((2025)). Verify phone numbers with the SMS Retriever API [Accedido el 2 de junio de 2025].
- Hosseini, A., & Rahmani, H. ((2021)). An Intelligent Recommender System for Medical Treatment using Data Mining Techniques. *Expert Systems with Applications*, 168, 114354.
- International Organization for Standardization. ((2021)). Health informatics — Security and privacy requirements of EHR systems [ISO 27799:2021]. <https://www.iso.org/standard/80303.html>
- JetBrains. ((2024)). Kotlin Coroutines Documentation [Consultado en junio de 2025]. <https://kotlinlang.org/docs/coroutines-overview.html>
- Jetpack KTX Team. ((2025)). core-ktx | Android Jetpack [Accedido el 2 de junio de 2025].
- Jurafsky, D., & Martin, J. ((2023)). *Speech and Language Processing (3rd ed., draft)* [Disponible en línea: <https://web.stanford.edu/~jurafsky/slp3/>]. Stanford University.
- Liu, X. e. a. ((2022)). ChatDoctor: A Medical Chat Model Fine-Tuned with Medical Domain Knowledge. *arXiv preprint arXiv:2210.02068*. <https://arxiv.org/pdf/2210.02068>

- LLC, G. ((2024)). *Firebase Documentation*. Consultado el 2 de junio de 2025, desde <https://firebase.google.com/docs>
- López, A., & Pérez, J. ((2022)). Sistema de gestión de citas médicas en Android con Firebase para una clínica en Lima.
- Manuel, L. T. G. ((2025)). Sistema experto para la recomendación y elaboración de planes alimenticios en base a condiciones físicas de los estudiantes de 1º año de secundaria en un colegio de Lambayeque. <https://tesis.usat.edu.pe/handle/20.500.12423/8492>
- Marisela, V. D. P. ((2022)). Sistema Recomendador de prescripciones médicas basado en clusterización. <https://dspace.utpl.edu.ec/handle/20.500.11962/29679>
- Martínez, L. ((2020)). Validación de datos críticos en apps médicas móviles. *Boletín Técnico de Informática Médica*, 5(2), 11-18.
- Matheus, L. D. A. ((2021)). Predicting Therapy Adherence : A Machine Learning approach - ProQuest. <https://www.proquest.com/openview/30afdf4850ba74cf714b89ccf3b6b113/1?cbl=18750&diss=y&pq-origsite=gscholar>
- Méndez, J., & Fernández, D. ((2022)). Eficiencia en el manejo de citas médicas usando Firestore en apps móviles. *Journal Latinoamericano de eHealth*, 8(3), 98-105.
- Nelly, H. J. A. ((2024), julio). Sistema de recomendación inteligente para mejorar la toma de decisiones en la fijación de precios de productos hortícolas en las microempresas. <https://repositorioacademico.upc.edu.pe/handle/10757/675796>
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. ((2022)). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf)
- Pérez Marín, D. ((2020)). *Sistemas conversacionales inteligentes: Diseño, implementación y aplicaciones*. Editorial Académica Española.
- Ramiro, D. R. M. E. ((2022), junio). Desarrollo de un sistema d recomendación basado en support vector machine (SVM) para el desarrollo del pensamiento computacional en estudiantes del nivel primario de educación básica regular ( EBR). <https://repositorio.ucsm.edu.pe/items/4a414c68-da1f-4c57-aefc-470be57fd20c>
- Rodríguez, D. ((2021)). Modelo de estados para control de citas médicas en apps Android.
- Romero, E., Jiménez, C., & Soto, J. ((2023)). Comparación de plataformas BaaS en el desarrollo de apps móviles: Firebase vs Amplify vs Supabase.
- Sebastián, R. H. I. ((2022), enero). Evaluación y aplicación de algoritmos de inteligencia artificial explicada para apoyar la toma de decisiones médicas en la salud fetal. <https://bibdigital.epn.edu.ec/handle/15000/22551>
- Stubblefield, J. W. ((2021)). Artificial Intelligence Algorithms for Medical Imaging and Healthcare - ProQuest. [https://www.proquest.com/openview/fb5b6b8e3c201eb3ab7a16c768f558/1?cbl=18750&diss=y&pq-origsite=gscholar](https://www.proquest.com/openview/fb5b6b8e3c201eb3ab7a16c768f558)
- Topol, E. ((2019)). *Deep medicine: how artificial intelligence can make healthcare human again*. Basic Books.
- Torres, K., & Ríos, A. ((2021)). Aplicación móvil para gestión de citas médicas utilizando Firebase y Android Studio para centros médicos del Callao.
- U.S. Department of Health and Human Services. ((2020)). Health Insurance Portability and Accountability Act of 1996 (HIPAA). <https://www.hhs.gov/hipaa/index.html>
- Vale, C. J. B. ((2021)). Evidence-Based Policy Decision Support System to Enhance In-Hospital Patient Experience in the State of Qatar - ProQuest. <https://www.proquest.com/openview/30afdf4850ba74cf714b89ccf3b6b113/1?cbl=18750&diss=y&pq-origsite=gscholar>

[proquest.com/openview/91b94c88984bd416edd5ac97ed4b1503/1?cbl=2026366&diss=y&pq-origsite=gscholar](https://proquest.com/openview/91b94c88984bd416edd5ac97ed4b1503/1?cbl=2026366&diss=y&pq-origsite=gscholar)

Vásquez, N., & León, H. ((2021)). Sistema móvil de gestión de pacientes para clínicas privadas con base en Firebase y Kotlin.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. ((2017)). Attention is all you need. *Advances in neural information processing systems*, 5998-6008. [https://arxiv.org/pdf/1706.03762](https://arxiv.org/pdf/1706.03762.pdf)

Víctor, Z. L. G. ((2023)). Aplicación web con machine learning para predecir el éxito de un tratamiento de anemia infantil de un centro de salud. <https://repositorio.usmp.edu.pe/handle/20.500.12727/13225>

Yamao, E., & Ernesto, B. A. A. ((2020)). App de recomendaciones alimentarias para reducir la mala alimentación en casos de anemia en niños del colegio "Apóstol de Punchauca". <https://hdl.handle.net/20.500.12727/6824>

Yesenia, A. T. J. ((2020)). Aplicación de inteligencia artificial en diagnósticos de enfermedades. [http://hdl.handle.net/20.500.12423/3378](https://hdl.handle.net/20.500.12423/3378)

Zhou, Y., Chen, X., & Liu, Z. ((2023)). Mixture of Medical Experts: Leveraging Submodel Specialization for Clinical Decision Support. *Journal of Artificial Intelligence in Medicine*, 145, 102-115. <https://doi.org/10.1016/j.artmed.2023.102115>